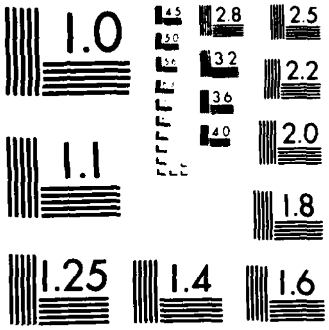
AD-A171 726 THE ENTITY-RELATIONSHIP APPROACH: A GOOD TOOL FOR TACTICAL DATA SYSTEMS?(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA R P KAUFFOLD 20 JUN 86 1/1

UNCLASSIFIED F/G 5/2 NL


1.0
4.5
5.0
5.6
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

2

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

THE ENTITY-RELATIONSHIP APPROACH:
A GOOD TOOL FOR TACTICAL DATA SYSTEMS?

by

Richard P. Kauffold

June 1986

Thesis Advisor: C. T. Wu

Approved for public release; distribution is unlimited.

AD-A171 724

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b. OFFICE SYMBOL (If applicable) 52
7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification) Unclassified
Entity-Relationship Approach: A Good Tool for Tactical Data System?

12. PERSONAL AUTHOR(S)
Richard P. Kauffold

13a. TYPE OF REPORT Masters Thesis	13b. TIME COVERED FROM ________ TO ________	14. DATE OF REPORT (Year, Month, Day) 1986 June 20	15. PAGE COUNT 63

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Entity-Relationship Model, Tactical Data System (TDS), Naval Tactical Data System (NTDS) Continued

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Entity-Relationship approach is investigated to determine its suitability for the construction of a logical database design for a tactical data system (TDS) to be used by surface ships in the U.S. Navy. Some motivation for the use of database techniques in the design of a TDS is given, and a conceptual schema design based on the Entity-Relationship approach is presented. This design includes Entity-Relationship diagrams in some detail for major entity and relationship sets for a TDS. An attempt to model behavior using Petri nets is described and several developed Petri nets are shown. It is concluded that the Entity-Relationship approach is workable for the task of building a TDS logical database design and that the resulting design if expressive and flexible.
(Continued)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL C.T. Wu	22b. TELEPHONE (Include Area Code) 408 646-3391 | 22c. OFFICE SYMBOL 52Wq

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Block # 18 (Continued)

Entity Set, Relationship Set, Tuple, Attribute, Petri Net

Block # 19 ABSTRACT (Continued)

It is also argued that the simplicity of the Entity-Relationship model makes design validation by real-world domain experts easier.

Approved for Public Release, Distribution Unlimited.

The Entity-Relationship Approach:
A Good Tool for Tactical Data Systems?

by

Richard P. Kauffold
Lieutenant, United States Navy
B.S., University of Nebraska at Lincoln, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1986

Author: ____________________________
Richard P. Kauffold

Approved by: ____________________________
C. Thomas Wu, Thesis Advisor

David K. Hsiao
____
David K. Hsiao, Second Reader

____
Vincent Y. Lum, Chairman
Department of Computer Science

Kneale T. Marshall
____
Kneale T. Marshall,
Dean of Information and Policy Sciences

ABSTRACT

The Entity-Relationship approach is investigated to determine its suitability for the construction of a logical database design for a tactical data system (TDS) to be used by surface ships in the U.S. Navy. Some motivation for the use of database techniques in the design of a TDS is given, and a conceptual schema design based on the Entity-Relationship approach is presented. This design includes Entity-Relationship diagrams in some detail for major entity and relationship sets for a TDS. An attempt to model behavior using Petri nets is described and several developed Petri nets are shown. It is concluded that the Entity-Relationship approach is workable for the task of building a TDS logical database design and that the resulting design is expressive and flexible. It is also argued that the simplicity of the Entity-Relationship model makes design validation by real-world domain experts easier.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Naval Ocean Systems Center (NOSC) in San Diego is currently engaged in research to determine the feasibility of employing database management techniques in the design of future tactical data systems. Of the many issues which need to be considered, the logical database design is one of the first. This thesis report presents the results of one effort to model a typical tactical data system (TDS) for Navy surface ships using the Entity-Relationship model proposed by Chen [Ref. 1].

The Entity-Relationship model (ER model) purports to provide the capability of modeling more of the semantics of real-world situations, and is the most widely understood of the new semantic data models. According to Chen [Ref. 2], it is an ideal model for use in the design of the conceptual (or enterprise) view as proposed by the ANSI/X3/SPARC report of 1975 [Ref. 3]. As Clemons points out [Ref. 4], the key feature of the ANSI/SPARC proposal is the use of a multi-schema architecture: one schema for the user's view (external), one schema for the enterprise view (conceptual) and one schema for the database management system's view (internal). Clemons discusses two claimed advantages for the ANSI/SPARC multi-schema architecture: ease of use, and enhanced data independence. The stability of the conceptual schema is a significant plus in that the enterprise view can evolve over time without fatal results to the other views. If the enterprise view can be systematically mapped to the internal view, the limitations of the database management system (DBMS) can be ignored when the enterprise view is designed. The process of using a model to design a conceptual schema is also referred to as building a logical database design.

Systematic mappings exist from schemas based on the ER model to those most commonly used for internal physical organizations so in that regard the ER approach can be used to design the conceptual (enterprise) view. Claims are made that the ER model allows more of the semantics of the real world to be expressed in the logical database design. It is the degree to which this is true that a model is good or bad when used for a particular database problem. Because there is no systematic way of constructing a logical database design, an evaluation of a model will be somewhat subjective, but the actual construction of a design is at least evidence that the model is workable for the given problem. Chapter III of this report contains one possible design for a typical TDS using the ER approach, and Chapter IV contains the conclusions drawn from this effort and proposes further research. Before the design is presented, however, Chapter II provides some motivation for applying DBM techniques to TDS systems.

II. MOTIVATION

The Naval Tactical Data System (NTDS) software is based on file-management techniques because database management techniques had not been invented when the system took form in the 1960's. Since that time, much DBMS work has been done and significant gains have been made in the organization of data. Because modern software engineering methods did not take root until the 1970's, the NTDS has little documentation and is difficult to understand and maintain. The need to modernize or replace NTDS has become more apparent, and DBMS techniques are being considered. Among the many issues that need to be addressed (i.e, speed, security, optimal hardware, DBMS type, etc.) is the question of what kind of logical database design will best suit the problem. Current literature contains proposals for many "semantic" data models, by which is meant data models that can express more of the real-world meaning found in situations to be modeled. The difficulty lies in the fact that many of these models are esoteric and therefore, despite their power, may not be useful for constructing logical database designs.

The entity-relationship (ER) approach proposed by Chen [Refs. 1 & 2] has the advantage of simplicity of concept yet it contains powerful semantical ability. The ER approach can be used to structure the logical organization of the data apropos of a domain in a way that captures more of the meaning of the data than conventional database models, but without producing complex and confusing designs. This is a significant advantage because database experts must rely on the real-world domain experts for the logical design of the database, and the ER approach provides a language to bridge conceptual gaps.

This brings the discussion to one of the most important goals of a logical database design: presentation of the conceptual (enterprise) view in a way that is understandable to the domain experts as well as the database experts. The process of creating a logical database design is not systematic: the choice of what portion of the real world to model is made by someone familiar with the domain being modeled. The only data desired in the model is useful data, which seems to go without saying, but who can best determine what data is useful? As a practical matter, the answer would appear to be the person knowledgeable of the real-world domain being modeled, i.e., the domain expert.

The domain expert needs an approach that is understandable and powerful, while the database expert needs a design that can be translated into the physical organization dictated by the DBMS. The problem is similar to that encountered by designers of expert systems. The designer of an expert system interviews domain experts, and from the knowledge gained, writes the rules for the system. These rules must be translatable into the particular language chosen (i.e., LISP, Prolog, etc). The difficulty lies in the fact that the domain expert may not have had training in predicate logic and therefore may not be able to confirm the rules as valid. In the database case, the ER approach seems to solve this type of problem by providing a common language for both the domain expert and the database designer, allowing the domain expert to confirm the validity of the logical design without detailed training in hierarchical, network or relational database management systems. The simplicity of the ER approach produces designs that lack ambiguity, yet are highly expressive.

In addition to having nice semantics, a good model must be flexible in that it must accommodate growth easily. Over time, more and more entities may be added to the logical design, and this shouldn't affect the internal or external views to the extent that they require complete revision. The ER approach can meet this requirement because logical designs constructed using the ER approach are not closed to additional entities and relationships as they are found to be necessary. Existing mapping algorithms can be used to update the internal physical organization of the data and the external application view.

Chen [Ref. 5] makes a strong case for the use of the ER approach in the logical database design process. He cites the advantages of understandability by non-database people, ease of the design process, and stability of the logical design (enterprise conceptual schema). The last advantage stems from the fact that the logical design does not have to be changed in order to change from one DBMS to another since it is independent of the DBMS used. He also points out that to change the user view (external schema) one wouldn't have to change the logical design, but simply re-map the enterprise conceptual schema to a new user schema. This flexibility seems to lend the ER approach to the logical database design for a TDS, which must continue to perform over several years in an evolving environment.

NTDS may be in use in the fleet for up to 30 years before it is replaced, and because the longevity of defense systems is increasing, its replacement may be operational for a much longer period. The next generation TDS must be flexible and have the ability to evolve and grow. The choice of data model for the logical database design of the next generation TDS will have an impact on combat readiness in the fleet for years after implementation. Hence, it seems justified to study the alternatives at this stage with care. The next chapter presents a logical database design for a TDS with a view to showing that a conceptual schema based on the ER approach is possible and has some semantic advantages in addition to the data-independence and ease-of-understanding advantages discussed in this chapter.

III. THE_DESIGN

A brief summary of Chen's model [Ref. 1] is in order before the TDS logical database design is presented. The ER approach uses the concepts of entity and relationship. Simply put, an entity is anything from the real world that can be thought of as a thing or concept and specifically identified in some way. Information from the real world can be characterized as entities or relationships among entities. An entity set is the set of all entities that meet some standard membership test, and a relationship set is a mathematical relation among entities. Both entities and relationships can have attributes, and they are defined as functions which map from entity sets or relationship sets into value sets or Cartesian products of value sets. Entities have keys, which are groups of attributes such that the mapping from the entity set to the corresponding value sets is one-to-one. One key is chosen to be the primary key, and the primary keys of entities associated with each other in a relationship can be taken together as the primary key of the relationship.

The ER diagrammatic technique uses rectangles to represent entity sets, elipses to represent attributes of entities, diamonds to represent relationships, arcs to connect those entities and relationships which are associated with each other and various kinds of arrows to connect attributes with entity sets or relationship sets. In this report, the standard method of indicating one-to-one, one-to-many, and many-to-many relationships is employed (i.e., using "1", "m" and "n" on the arcs between rectangles and diamonds). Arrows, -->, are used to connect entity sets or relationship sets and their many-to-one attributes; double-sided arrows, <-->, are used for one-to-one attributes; double arrows, -->>, are used for many-to-many multivalued attributes; double-sided double arrows, <-->>, are used for one-to-many multivalued attributes. The primary key is identified by underlining the name of the attribute(s).

The process of designing a logical database is by nature an iterative one, and the design presented here is no exception. Some of the original ideas have survived to this stage and some have been discarded and replaced with newer ones. The intention is to show that a logical database design for a TDS can be completed using the ER approach, and no claim is made that this is the best possible logical database design for a TDS. The process of categorization seems to be particularly individualistic and different domain observers will categorize entities in different ways based on their knowledge, experience and biases. The important question here should not be "is this a good design?", but rather "is the ER approach a worthwhile tool for TDS logical database designers?"

In section A of this chapter, the basic entities and their attributes are described and section B discusses the relationships between entities. Section C introduces some additional entities and relationships and incorporates them in the overall schema. The final section discusses a proposed technique to model the behavior of entity and relationship sets, which could be important for a TDS.

A. THE ENTITIES

In one view, the most important entities involved in a surface ship battle group tactical picture are contacts, sensors and weapons. It is with these three concepts that we begin the development of the logical database design for a TDS.

A contact is an object that is sensed by the ship; it can be in the air, on the surface or under the surface. Figure 3.1 shows the ER diagram (ERD) of the entity set called CONTACT. The ISA relationships are to indicate the decomposition of CONTACT into three sub-entities and the design was made this way because there can be small variations in the attribute sets of the three entities. Figures 3.2, 3.3, and 3.4 shows the ERDs for each of the sub-entities with typical attributes, and Table 3.1 lists the attribute domains (value sets) for CONTACT.

The primary key for CONTACT is "Track #" which is an artificial attribute. Its value is assigned as soon as an

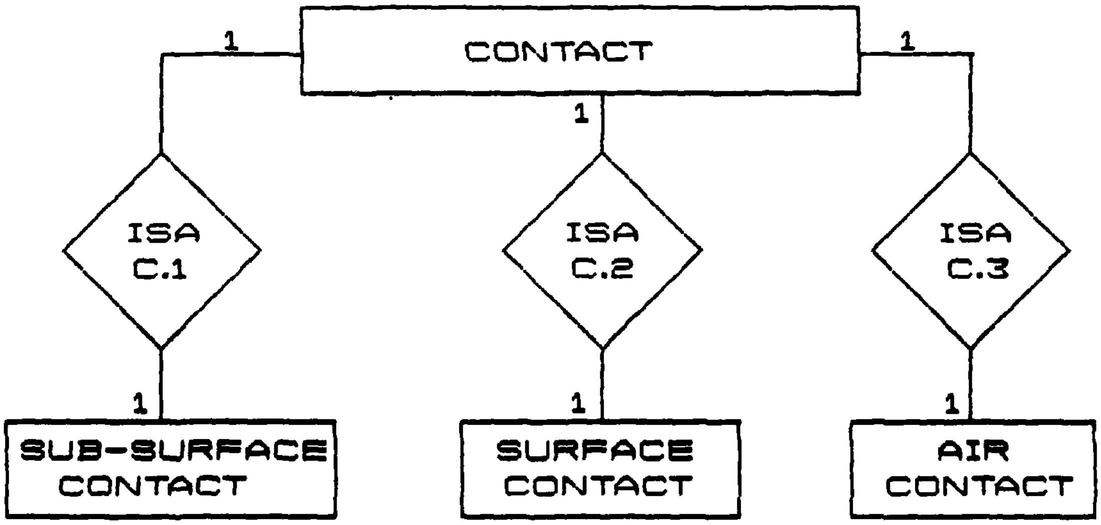


Figure 3.1   ERD for CONTACT

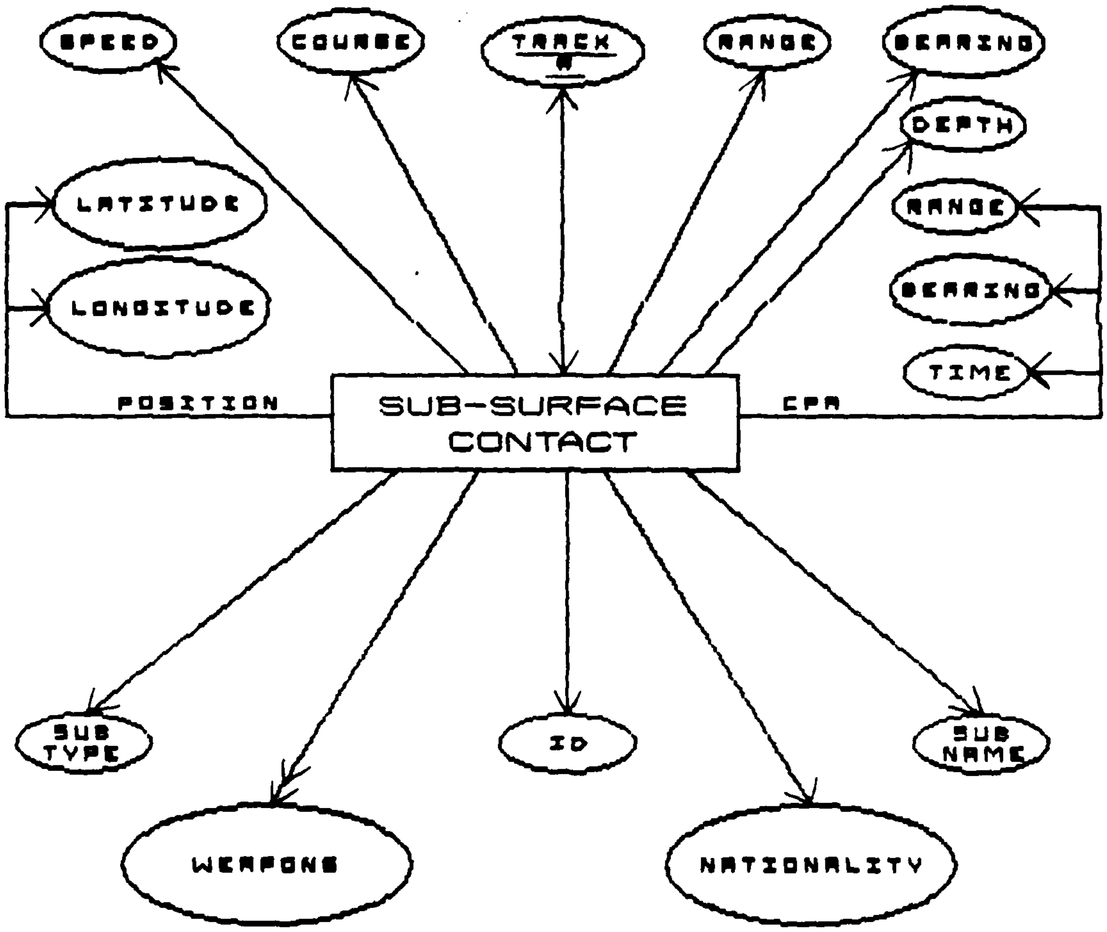


Figure 3.2 ERD for SUB-SURFACE CONTACT

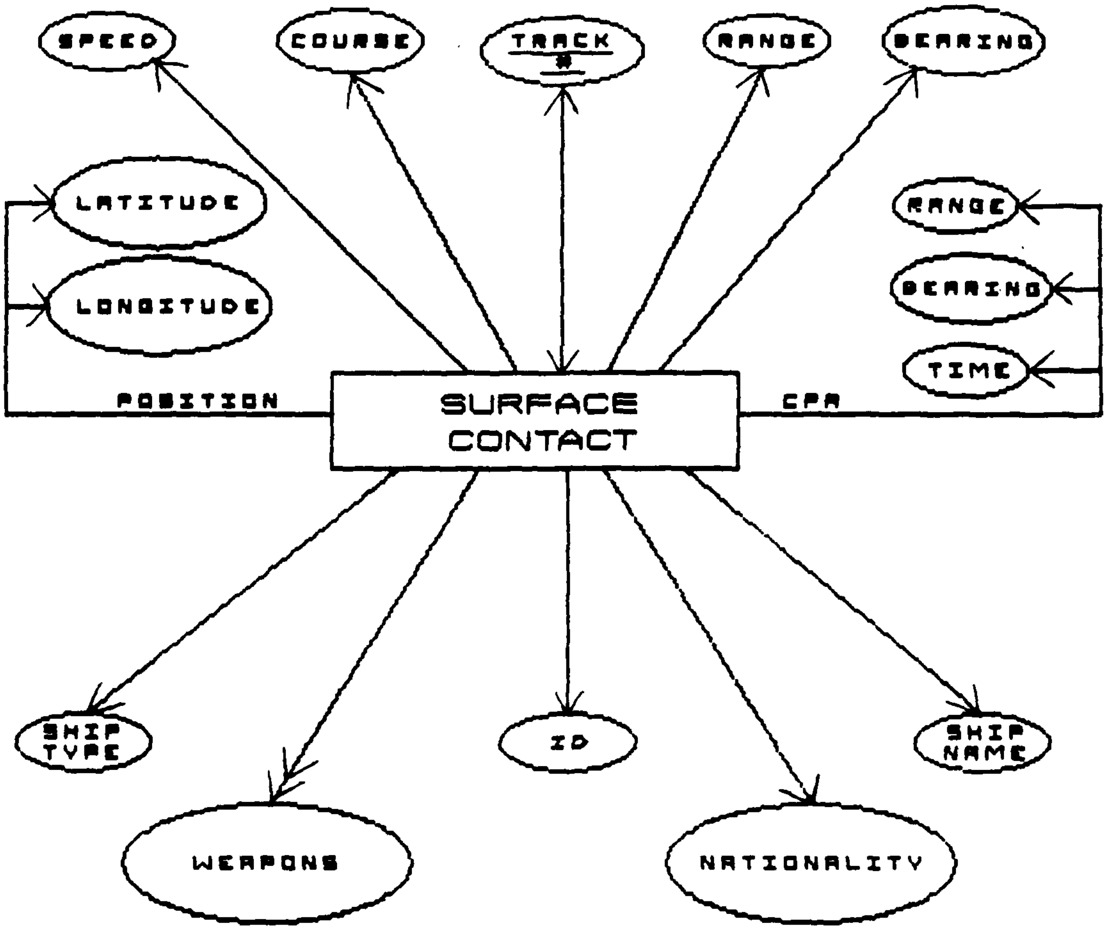


Figure 3.3 ERD for SURFACE CONTACT

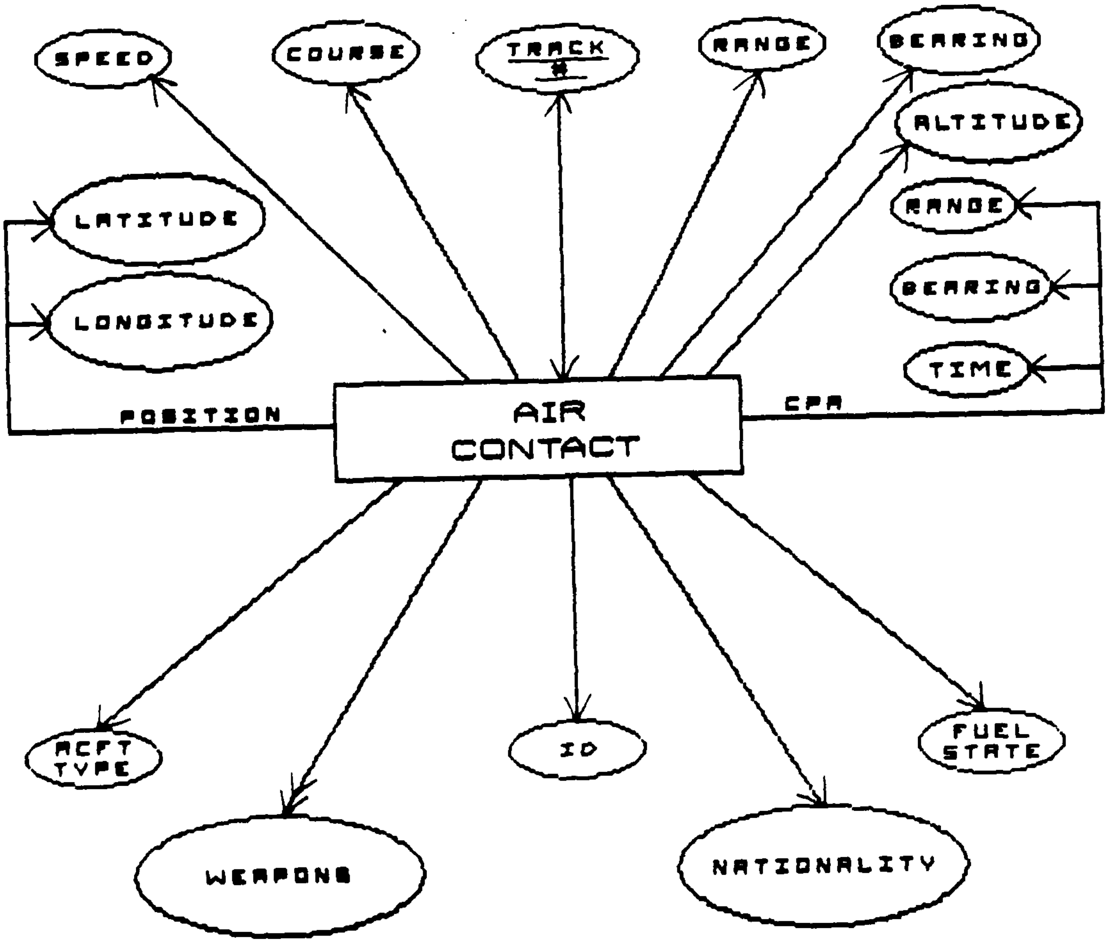


Figure 3.4 ERD for AIR CONTACT

TRACK #	: (0000,0001,0002, ... ,9999)
COURSE	: (000,001,002, ... ,359) (degrees True)
SPEED	: (0000,0001,0002, ... ,9999) (knots)
RANGE	: (0000,0001,0002, ... ,9999) (thousands of yards)
BEARING	: (000,001,002, ... ,359) (degrees True)
LATITUDE	: (00:00,00:01N, ... ,00:59N,01:00N,01:01N, ... , 90:00N,00:01S, ... ,90:00S) (degrees and minutes of arc)
LONGITUDE	: (000:00,000:01E, ... ,000:59E,001:00E, ... , 180:00,000:01W, ... ,180:00) (degrees and minutes of arc)
POSITION	: (LATITUDE X LONGITUDE)
TIME	: (0000:00,0000:01, ... ,0000:59,0001:00, ... , 0059:59,0100:00, ... ,2359:59) (hours, minutes and seconds of clock time)
CPA	: (RANGE X BEARING X TIME)
ID	: ("friendly", "hostile","unknown")
WEAPONS	: ("guns","torpedos","AAW missiles","ASUW missiles", "ballistic missiles", etc.)
SUB_TYPE	: ("conventional fast attack", "nuclear fast attack", "conventional ballistic missile", "nuclear ballistic missile")
SHIP_TYPE	: ("merchant","patrol craft","frigate","destroyer","cruiser","battleship","aircraft carrier","replenishment","repair")
ACFT_TYPE	: ("land based bomber","sea based bomber", "land based patrol","sea based patrol", "fighter","early warning","ECM(jammer)", "reconnaissance","rotary-wing", "civilian(commercial)","civilian(private)", "cruise missile","AAW missile")
NATIONALITY	: ("name":"name" represents a politically sovereign state) (e.g., "Italy")
SUB_NAME	: ("name":"name" represents an individual submarine) (e.g., "USS Omaha")
SHIP_NAME	: ("name":"name" represents an individual ship) (e.g., "HMS Invincible")
FUEL_STATE	: (00,01, ...,99) (percentage of fuel left onboard -- "friendly" aircraft only)
DEPTH	: (00,01, ... ,99) (hundreds of feet)
ALTITUDE	: (00,01, ... ,99) (thousands of feet)

Table 3.1 "CONTACT" ATTRIBUTE DOMAINS (VALUE SETS)

entity tuple is established (as when a contact is first sensed by a sensor). RANGE and BEARING are attributes that give the contact's position relative to the platform from which the view is taken (usually called "own ship"), while POSITION is a composite attribute that gives location relative to the earth's latitude and longitude. CPA is a composite attribute that gives the range, bearing and time of the "closest point of approach" of the contact to own ship. COURSE, SPEED, DEPTH, ALTITUDE, and NATIONALITY are self-explanatory attributes. ID would have one of three values: "friendly", "hostile" or "unknown". WEAPONS is the only multi-valued attribute, and has values that represent different types of weapon systems. TYPE has values that represent things like "destroyer", "aircraft carrier", and "replenishment" for surface contacts, "nuclear fast attack" and "conventional ballistic missile" for submarine contacts, and "fighter", "sea based patrol" and "cruise missile" for air contacts (see Table 3.1 for a more complete listing).

The value sets of each attribute can be easily changed as necessary without significant impact on the overall logical database design. In fact, attributes can be added or deleted without trouble. A logical design that would model a more robust view of CONTACT would not have to be structurally different from the one presented here, a statement that is equally true when said about the SENSOR and WEAPON entity sets.

A sensor is an object that is designed to develop data about contacts and to provide this data to weapon systems. Figure 3.5 shows the ERD for SENSOR. For a surface ship, there are six primary sensors which are shown in the diagram, but more sub-entities can be added to the basic entity set SENSOR as they are developed and implemented in the fleet. MAD (magnetic anomaly detection) is carried by certain types of aircraft and can find submarines hidden beneath the surface of the water. Figure 3.6 shows the ERD for MAD, including the probable attributes. The ISA relationships are numbered in a way that allows them to be distinguished from other ISA relationships. For example, the "ISA S.1.1" relationship means that it is the first sub-entity of the first sub-entity of SENSOR (the "S" part). Although the attributes are shown to be the same for both ROTARY WING MAD (helicopter MAD) and FIXED WING MAD, it is feasible that each would have additional attributes peculiar to the aircraft. The key for all SENSOR entities would be SENSOR # and the attribute domains for all SENSOR attributes can be found in Table 3.2.

The ERD for the SONAR entity set is shown in Figure 3.7 (the SENSOR attributes are not shown because they are the same for all six SENSOR sub-entity sets). The ISA relationships are numbered in the same manner as described above for MAD, but in SONAR, there are two more levels. This demonstrates the flexibility of the use of ISA relationships

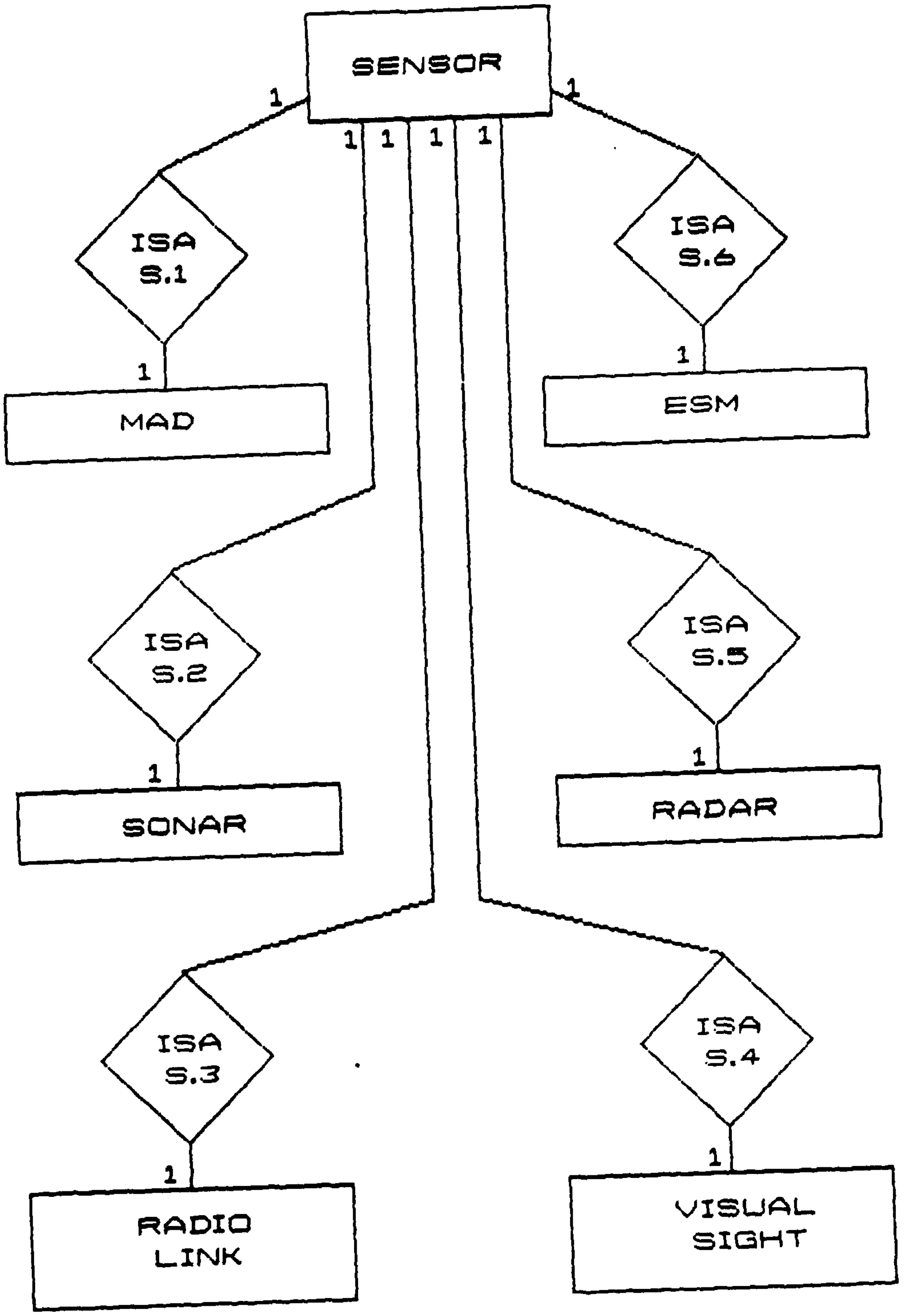


Figure 3.5 ERD for SENSOR

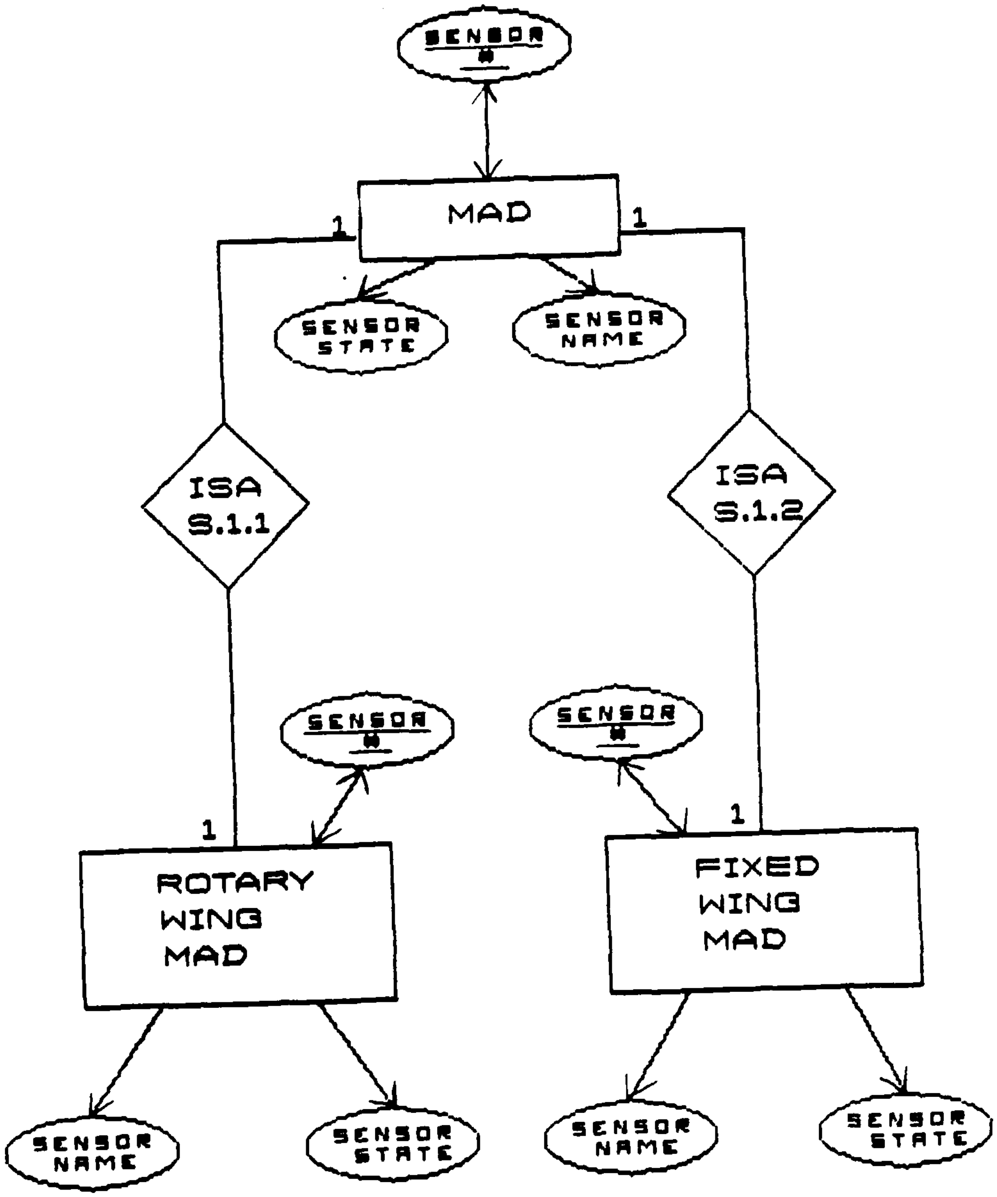


Figure 3.6 ERD for MAD

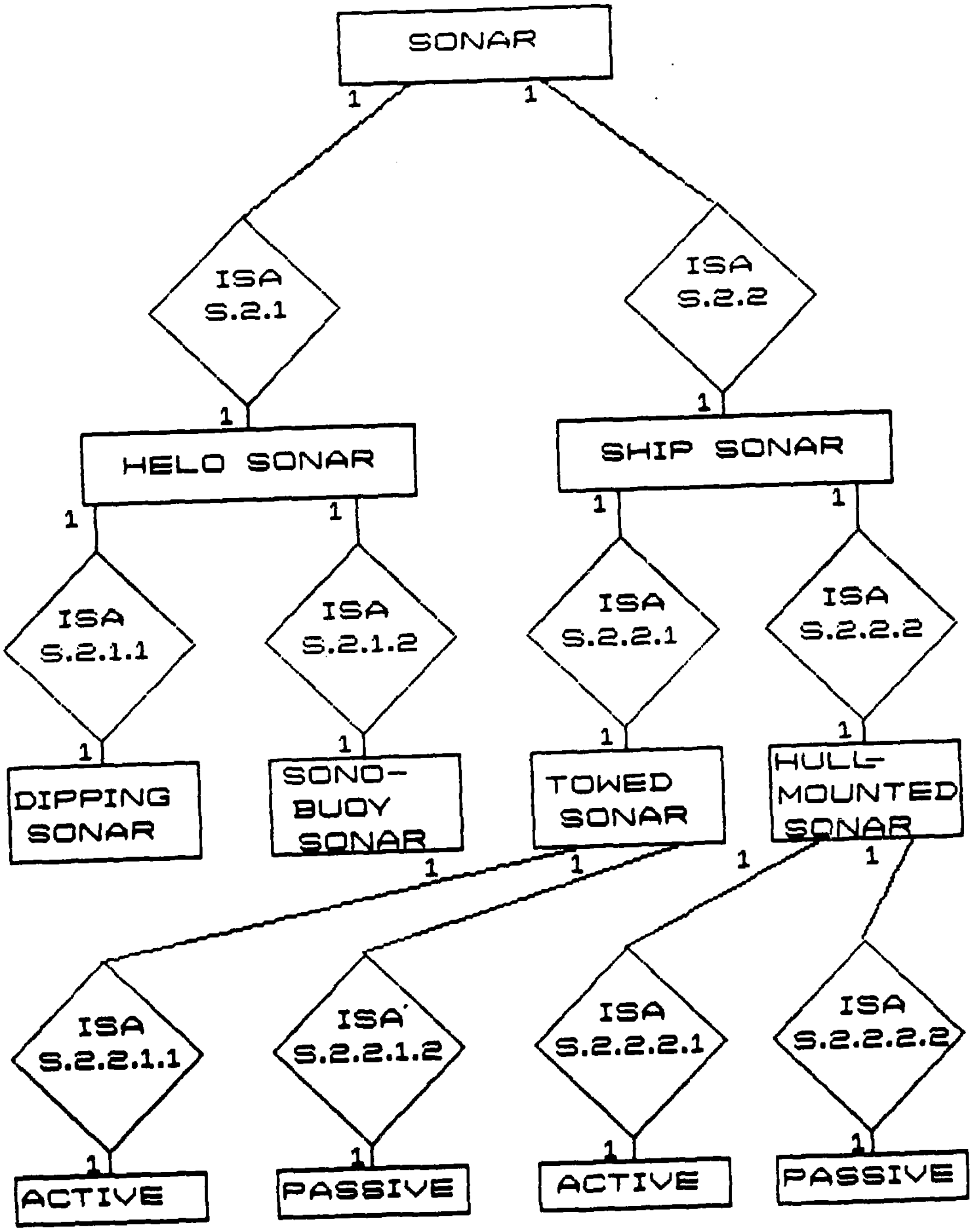


Figure 3.7 ERD for SONAR

to model the semantics of real world situations that are naturally hierarchical in organization. The basic division of SONAR into HELO SONAR and SHIP SONAR is because most ships using a TDS would have sonar input from both own ship and from one or more helicopters assigned to the ship. Further breakdown of HELO SONAR is because some helo sonars are submerged into the water via a cable from the helicopter and some helo sonar information comes from sonobuoys which are dropped into the water and transmit data via radio frequencies. SHIP SONAR is either towed by the ship or mounted on the hull of the ship, and either type can be active (radiating sound and listening for returns from contacts) or passive (listening for machinery noises to find contacts).

RADIO LINK has no sub-entity sets because it is the conceptual sensor from which data is obtained from other platforms (other ships of the battle group and friendly aircraft not assigned to own ship). Contacts that are sensed by the sensor RADIO LINK are remote (as opposed to local), because data originates from remote sensors (i.e., those on other ships). The remote sensors are all members of one of the other SENSOR sub-entity sets shown in Figure 3.5. It is convenient to have a sub-entity set of SENSOR devoted to remote information so that there will be no confusion between sonars on one destoyer in the battle group and sonars on another. Often, a commander will put more faith in data originating from one source than from another due to known equipment differences or other anomalies.

Figures 3.8 and 3.9 show the ERDs for VISUAL SIGHT and ESM (electronic surveillance measures). These are relatively simple sub-entities of SENSOR that can be found on the ship and also on the helicopter assigned to the ship. VISUAL SIGHT may seem like an obsolete choice for a sensor in the modern technological age, but it is important because it is often necessary to have a correlating visual sighting of a contact in order to have a high level of confidence in the contact data. ESM is a sensor that searches for electromagnetic radiation (radio signals, radar signals etc.) and the data developed by ESM can be used to identify contacts, or at least narrow the possibilities.

The ERD for RADAR is found in Figure 3.10, and shows four sub-entity sets. Fire Control Radars are primarily used to control the firing of weapon systems but they also can provide information to a TDS. Other radars are classified as either surface search (to indicate that they are directed toward contacts on the surface of the water), or air search (to indicate that they are directed to look for air contacts).

Figure 3.11 presents the ERD for WEAPON and indicates that a weapon is one of three types: ASW (anti-submarine warfare), ASUW (anti-surface warfare), or AAW (anti-air warfare). In this classification, there are three different

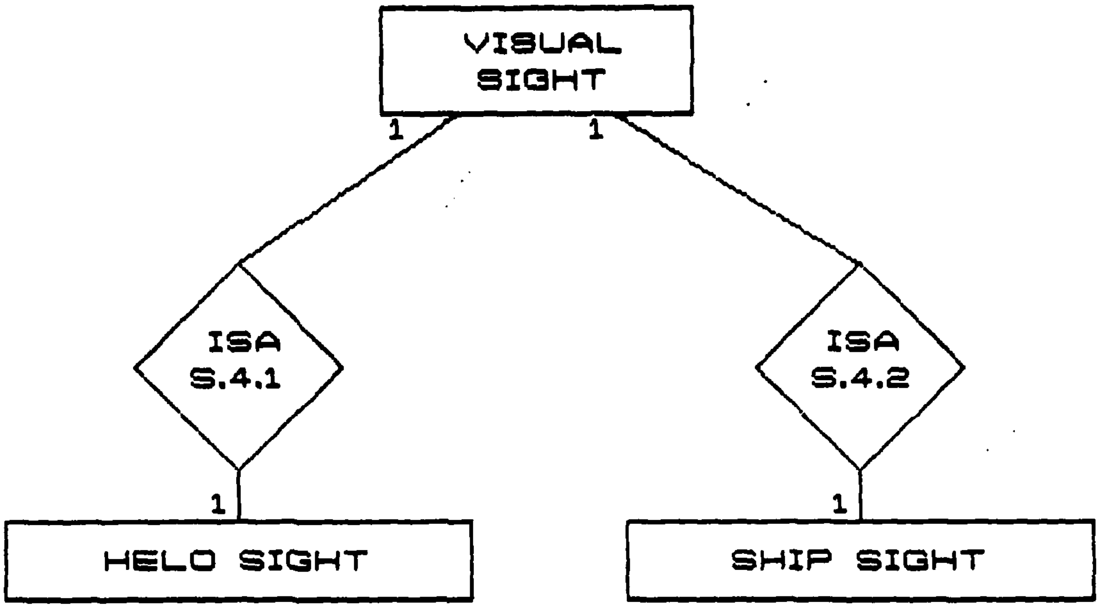


Figure 3.8 ERD for VISUAL SIGHT

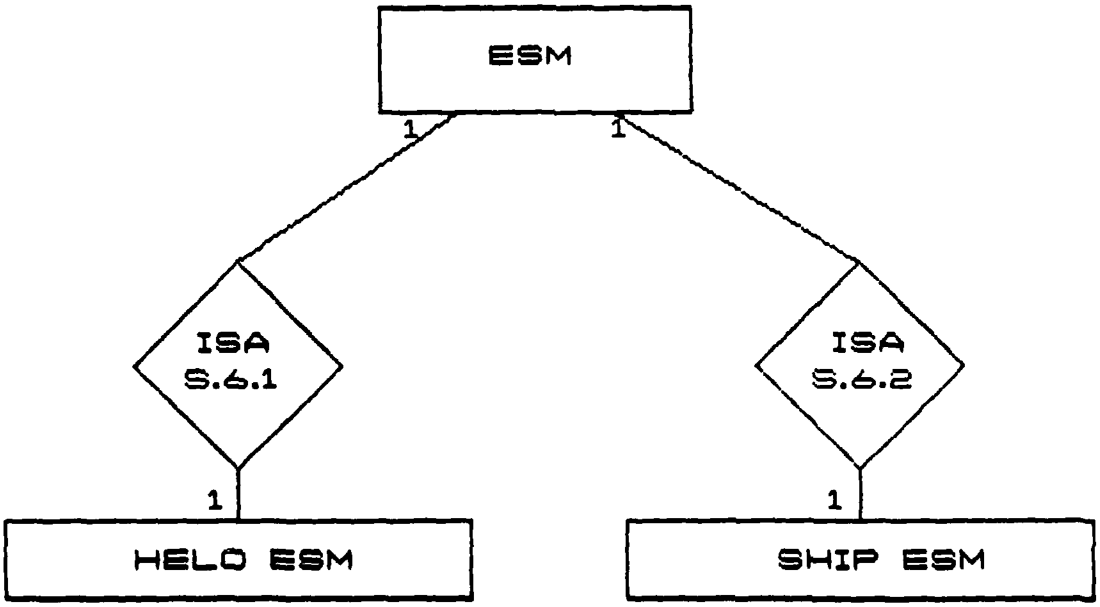


Figure 3.9 ERD for ESM

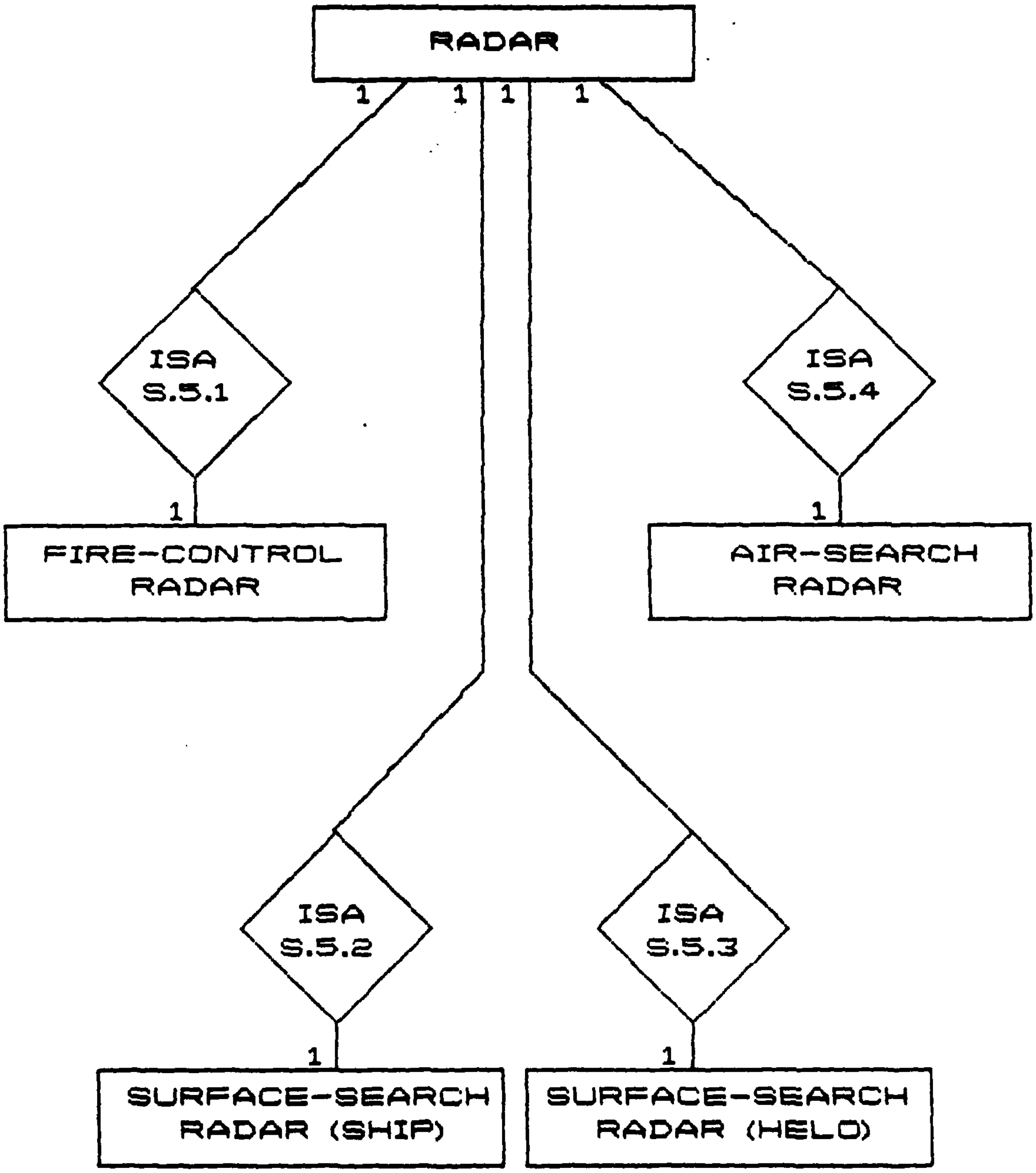


Figure 3.10 ERD for RADAR

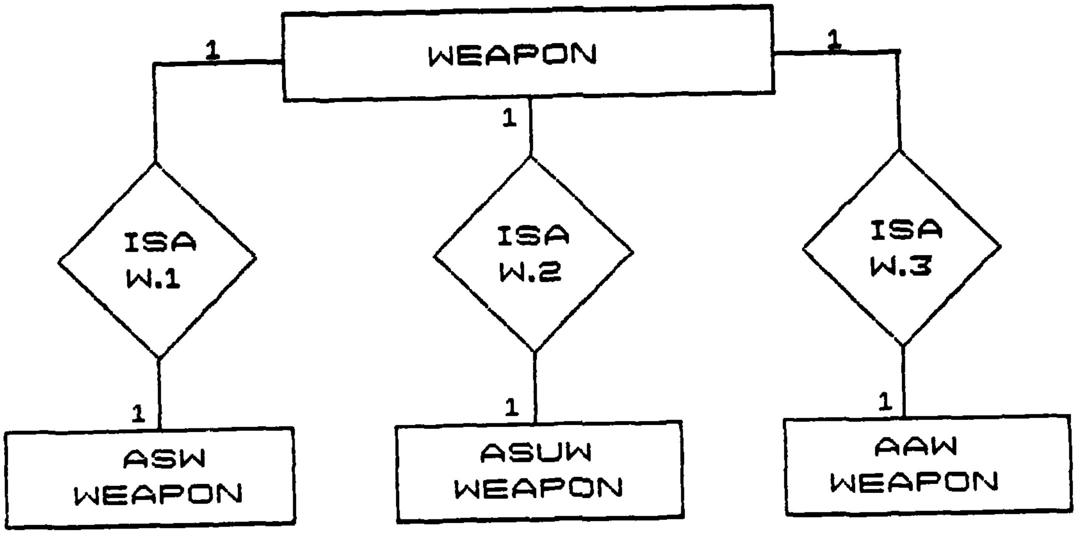


Figure 3.11 ERD for WEAPON (1)

```
SENSOR #       : ( Axxx : [ A= M(Mad)|S(Sonar)|L(Radio Link)|
                                E(ESM)|V(Visual)|R(Radar) ]
                            ^ x ∈ {0,1, ... 9} }
SENSOR NAME    : ( "name":"name" is a string representing a
                   particular sensor designation )
STATE          : ( "OOC"(out of commission),"OOS"(out of
                   service),"STBY"(standby),"ENERGIZED",
                   "TRACKING" )
```

Table 3.2 "SENSOR" ATTRIBUTE DOMAINS (VALUE SETS)

```
WEAPON #       : ( ABxxx : [ A= X(for ASW)|Y(for ASUW)|
                             Z(for AAW) ]
                           ^ [ B= O(for own ship)|H(for helo) ]
                           ^ [ x ∈ {0,1,2, ... ,9} }
WEAPON NAME    : ( "name":"name" represents a particular weapon
                   system ) (e.g.,"MK 46 Torpedo","ASROC",
                   "NATO Seasparrow Missile", etc.)
TYPE           : ( "warshot","exerciseshot" )
STATE          : ( "OOC"(out of commission),"OOS"(out of
                   service),"STBY"(standby),"ENERGIZED",
                   "SEARCHING","TRACKING" )
```

Table 3.3 "WEAPON" ATTRIBUTE DOMAINS (VALUE SETS) types of weapons because there are three different types of contacts (targets). Each tuple representing an entity in the WEAPON entity set will have values for four attributes. The attributes are not shown in the WEAPON figures because they are all the same, but the attribute value sets can be found in Table 3.3.

Figures 3.12, 3.13 and 3.14 show the ERDs for ASW WEAPON, ASUW WEAPON and AAW WEAPON respectively. Torpedoes turn out to be the most complex weapon from this classification standpoint because they can be delivered in many ways and it is important to the tactical picture to distinguish among those delivery methods. There are three types of missiles in use today: cruise missiles used against surface targets, guided missiles used against air targets which threaten the battle group, and point defense missiles used against air targets that threaten own ship. GUNS is an interesting sub-entity set because it can be found in all three main sub-entity sets of WEAPON, namely ASW WEAPON, ASUW WEAPON, and AAW WEAPON. This can be modeled in an overall schema for WEAPON, such as the one shown in Figure 3.15.

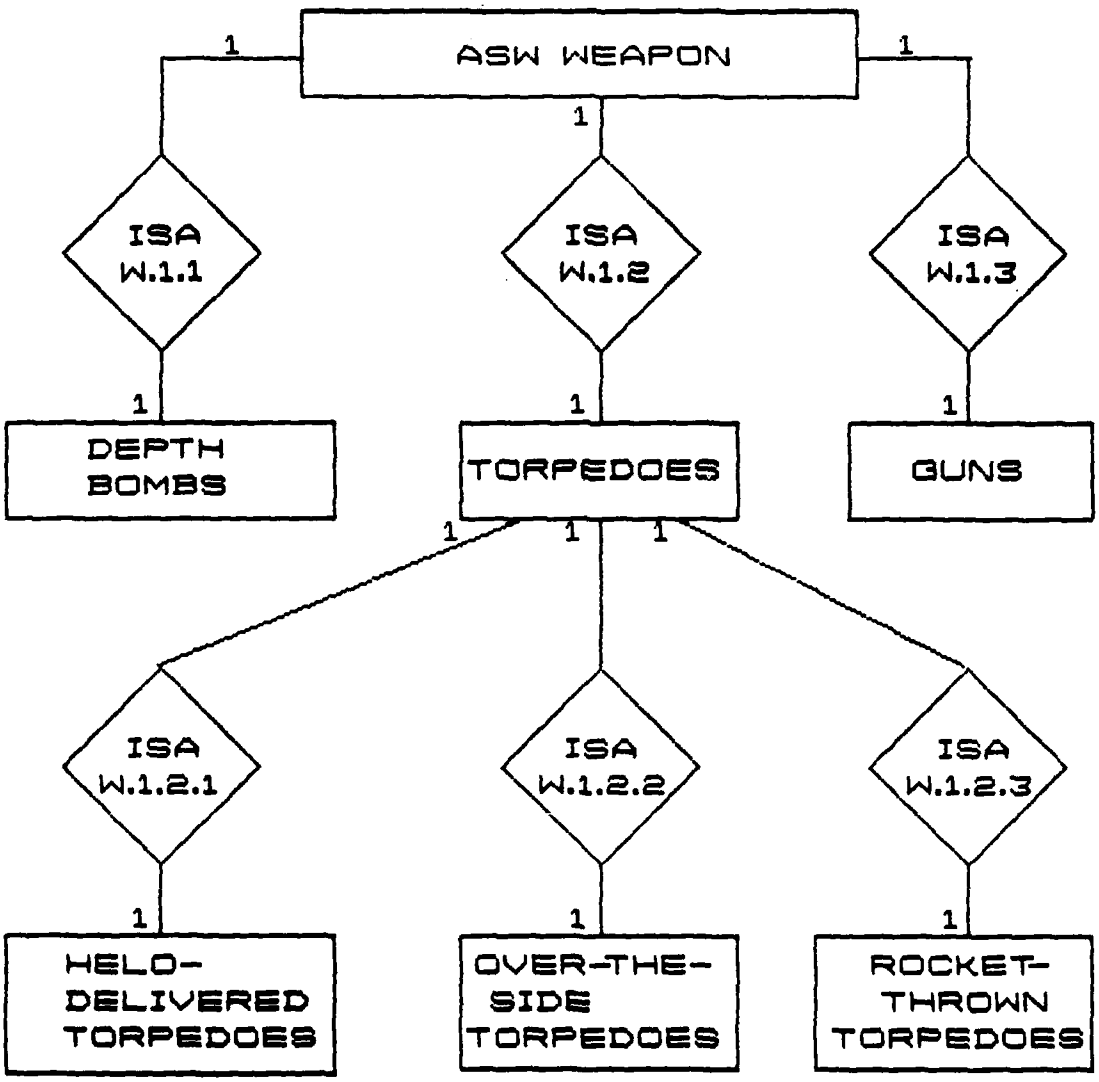


Figure 3.12 ERD for ASW WEAPON

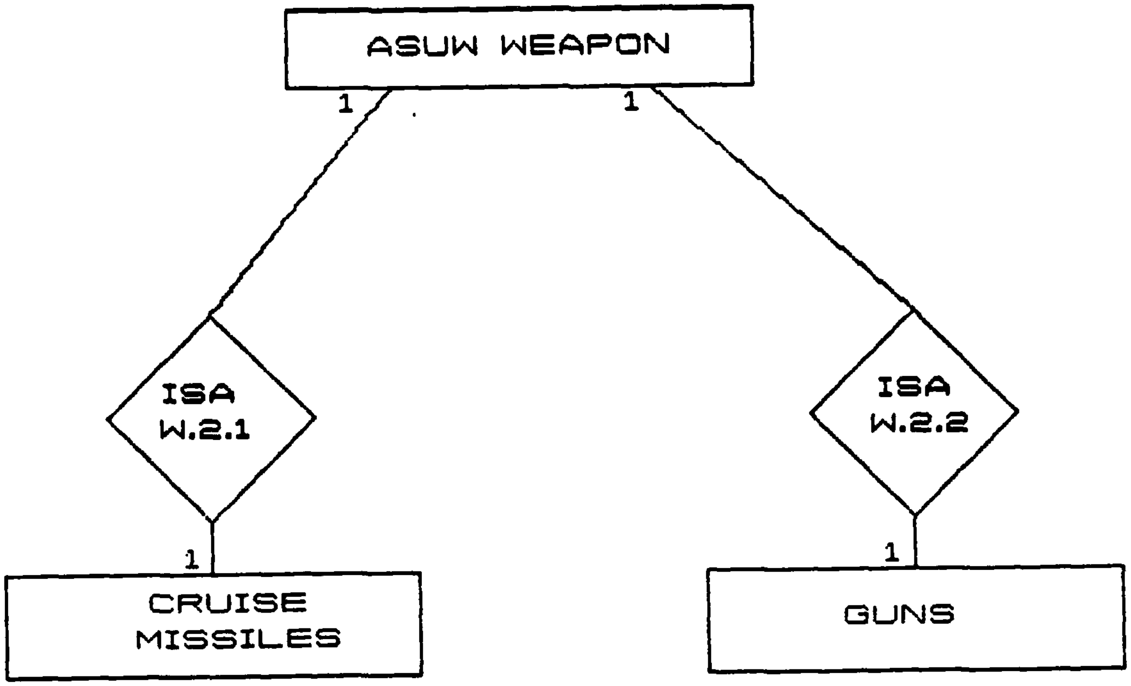


Figure 3.13 ERD for ASUW WEAPON

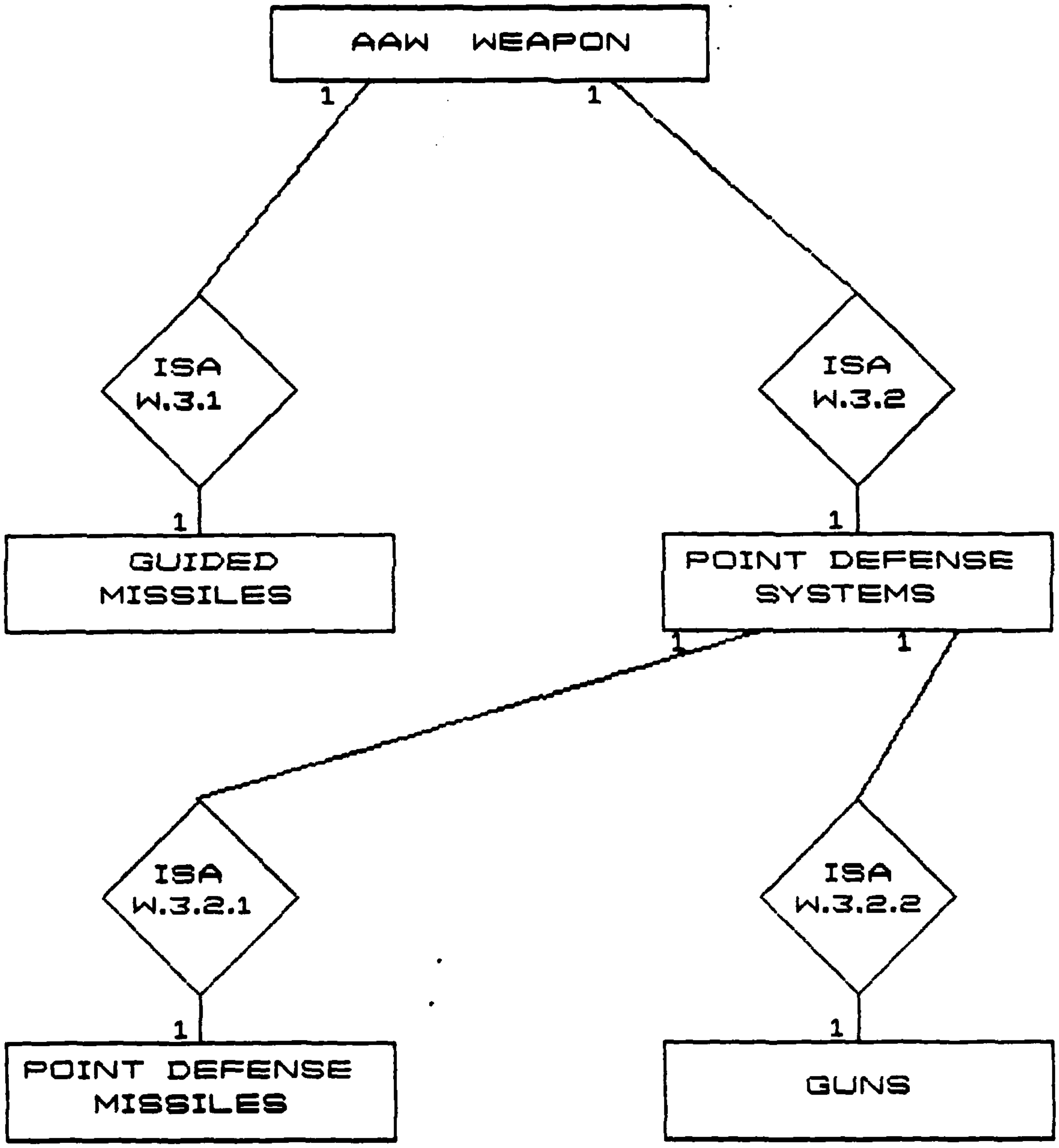


Figure 3.14 ERD for AAW WEAPON

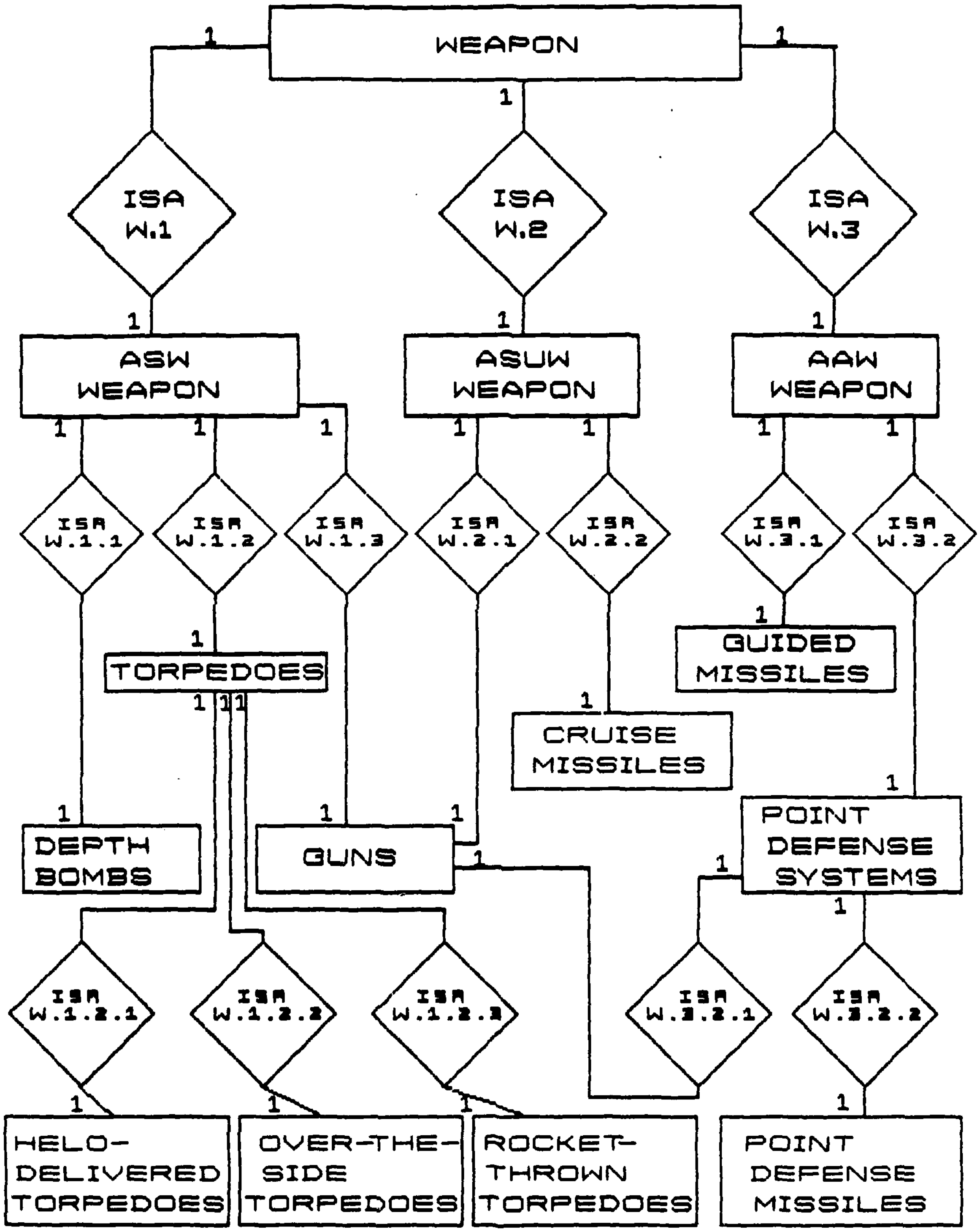


Figure 3.15 ERD for WEAPON (2)

B. BASIC RELATIONSHIPS

The three basic entity sets can be related by three basic relationship sets which are depicted in Figure 3.16. Each entity set has a many-to-many relationship with the other two entity sets and these three relationships will be discussed in turn.

First, there is an association between contacts and sensors, because each contact may be sensed by one or more sensors and each sensor may sense one or more contacts. This association is embodied in the relationship set DETECTION. The primary key for DETECTION is the composite of the keys for CONTACT and SENSOR, namely, "Track_#" and "Sensor_#". Tuples found in DETECTION would have values for the attributes of the associated contact and sensor (some could be nil). No additional attributes seem required, but it may be desirable for one reason or another to give DETECTION attributes of its own. (None of the basic relationships were given attributes in this study.)

SENSOR is associated with WEAPON because sensors provide weapon systems with data about contacts. Each sensor may direct one or more weapon systems toward contacts, and each weapon system may be directed by one or more sensors toward contacts. The name chosen to represent this association is DIRECTION. Once again, the primary key will be the composite of the primary keys of the entity sets that are related, and no additional attributes were deemed necessary.

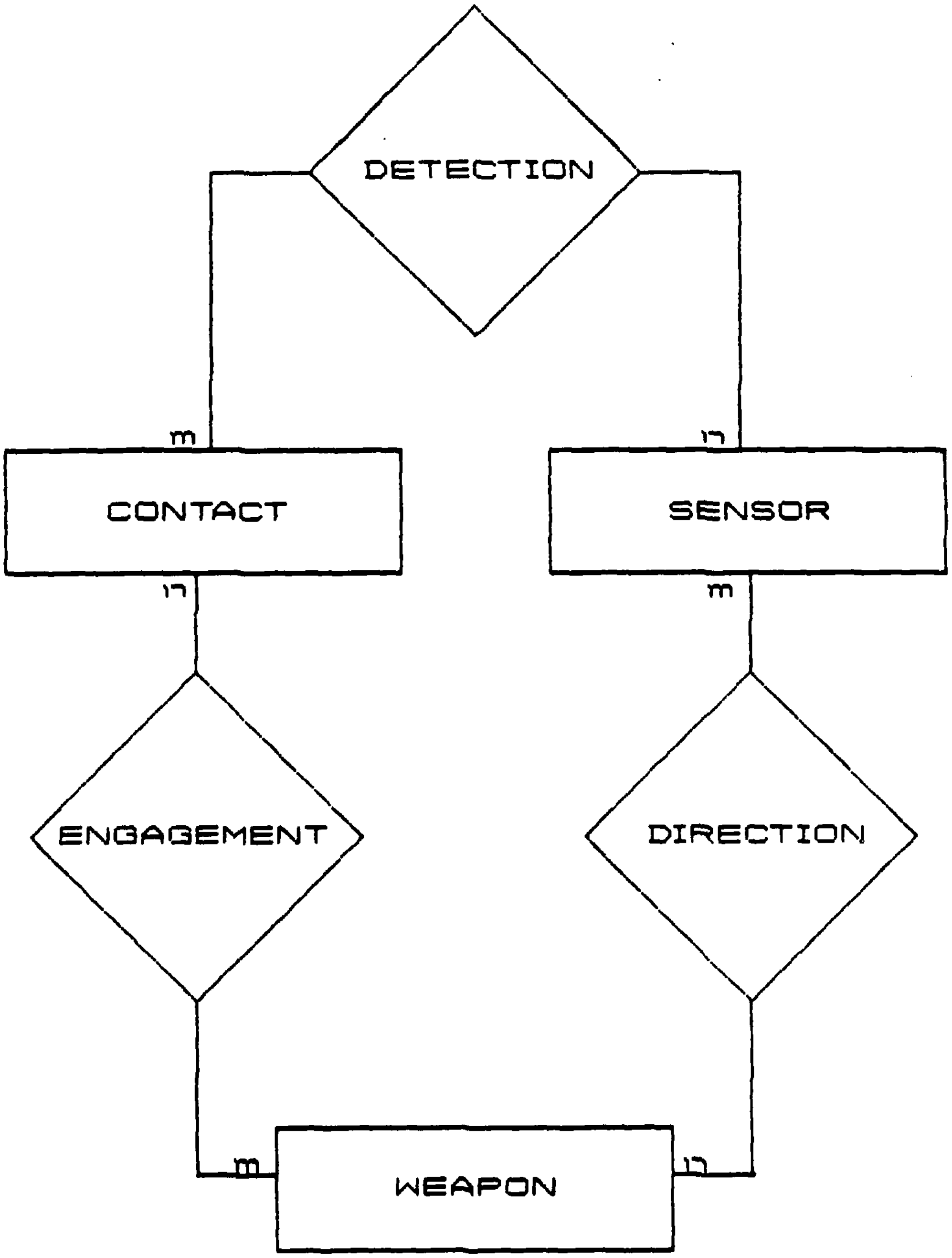


Figure 3.16 The Basic Relationships

In actual combat systems today, weapons systems often have integral sensors which control and guide the weapons, but these sensors are being considered members of the SENSOR entity set because they share the SENSOR attributes and associations. Tuples found in DIRECTION, would have values for the attributes of the associated weapon system and sensor.

Finally, the third of the basic relationship sets is ENGAGEMENT, and it embodies the association between contacts and weapons systems. When a commander orders the engagement of a target (contact) by a weapon system (weapon), this association is formed, and tuples found in ENGAGEMENT provide data about the associated contacts and weapons systems. Obviously, ENGAGEMENT only rarely contains tuples, but it is certainly the basic relationship between a weapon system and a contact.

C. THE BIG PICTURE

Before the overall conceptual schema is presented, three new entity sets will be introduced. One, COMMANDER, is needed to model the control of weapon systems, and two others, OWN SHIP, and ENVIRONMENTAL CONDITIONS, are needed to model limitations on sensors and weapon systems that change over time.

The COMMANDER entity set is decomposed into three sub-entity sets: ASW COMMANDER, ASUW COMMANDER and AAW COMMANDER because in many tactical situations it is common to have one commander for each warfare type. COMMANDER follows the pattern of CONTACT and WEAPON in that all three entity sets are sub-classified by their location environment (sub-surface, surface or air). Figure 3.17 shows the COMMANDER entity set, the three sub-entity sets and the probable attributes. "Call_Sign" is an alphanumeric string that specifically identifies each commander, and "Location" is a string that indicates in which platform the commander is embarked. The attribute domains would be suitably constructed to provide for these values. Figure 3.18 shows COMMANDER incorporated in the previously developed schema through the use of the relationship set CONTROL. Each commander may have control of many weapon systems, but each weapon system is controlled by only one commander, and so CONTROL is a one-to-many relationship as indicated.

The final two entity sets presented are special cases, because they each contain one and only one entity (or tuple representing the entity). Figures 3.19 and 3.20 show the entity sets OWN SHIP and ENVIRONMENTAL CONDITIONS respectively along with some potential attributes. OWN SHIP models the data peculiar to own ship and would be updated at some periodic interval, for example every minute or so. Course is important because some sensors are masked ahead or astern because of their location on the ship and because weapons systems can also be masked. Since contact bearings

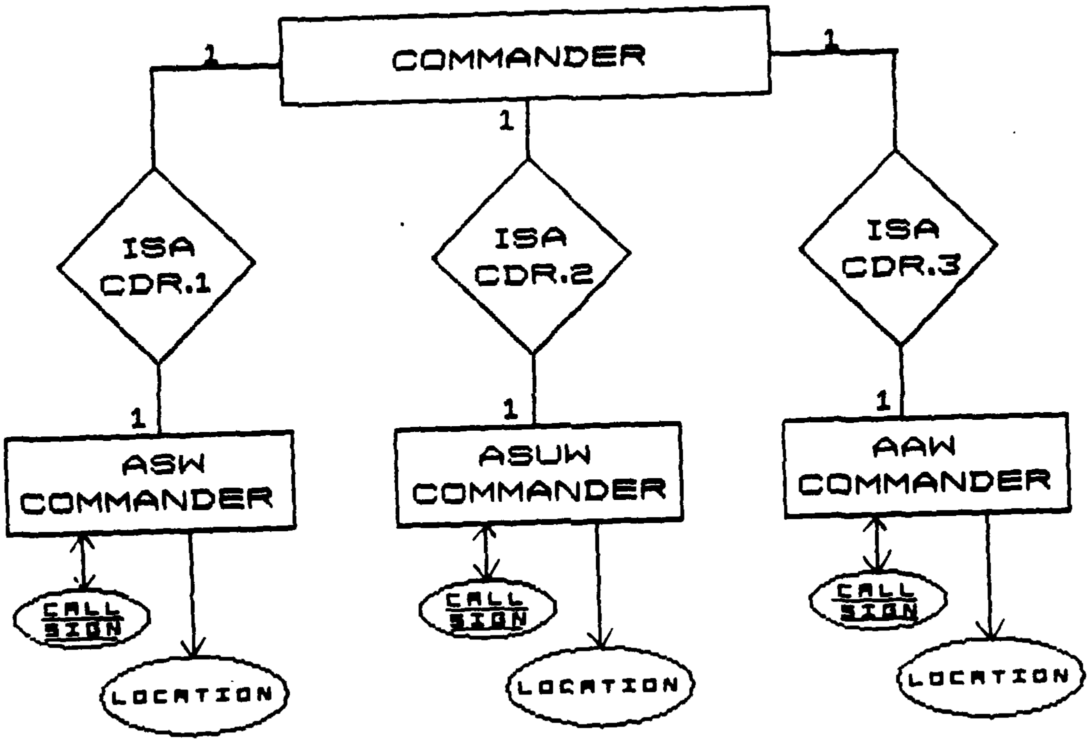


Figure 3.17 ERD for COMMANDER

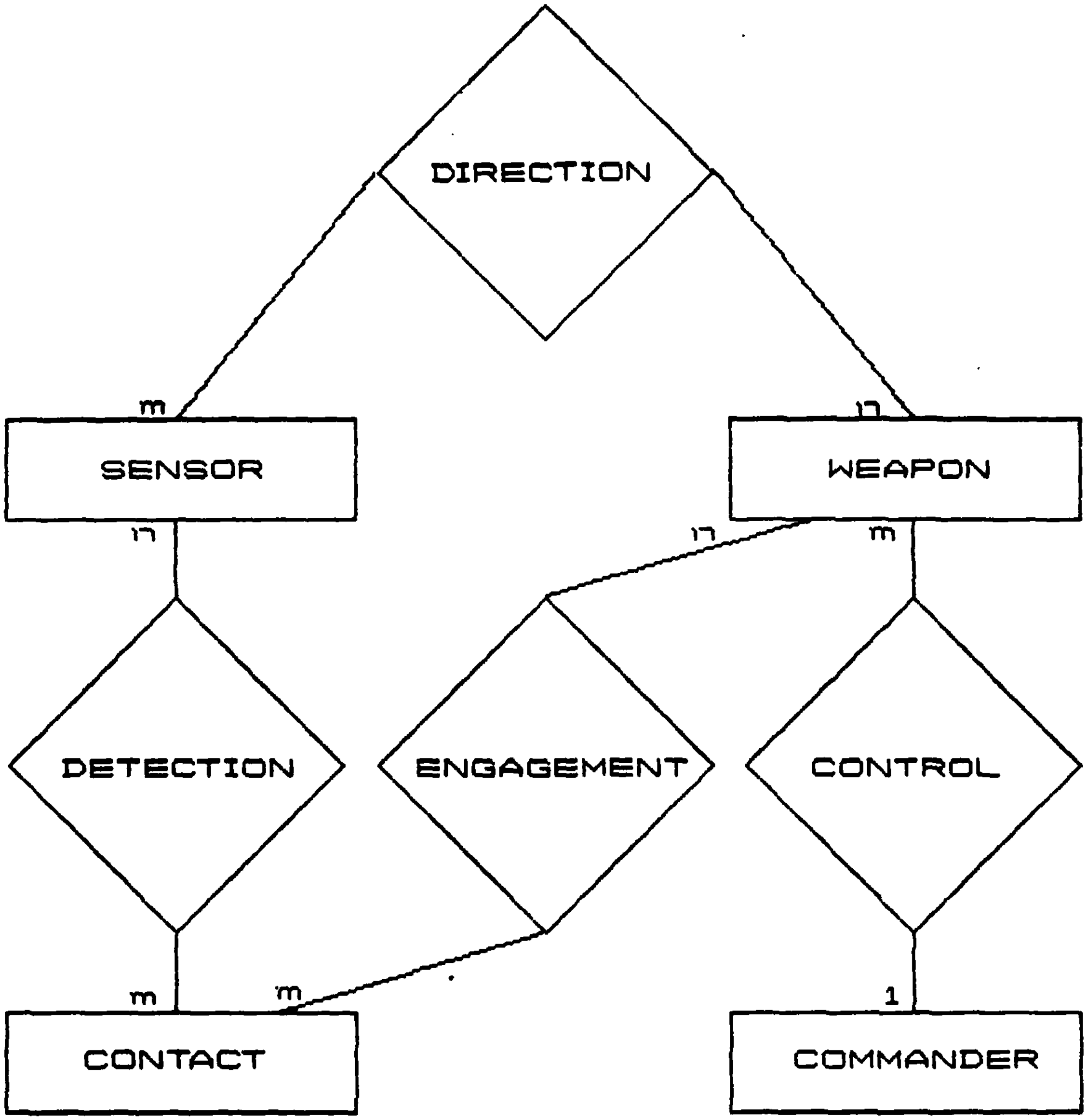


Figure 3.18 ERD Showing Schema With COMMANDER

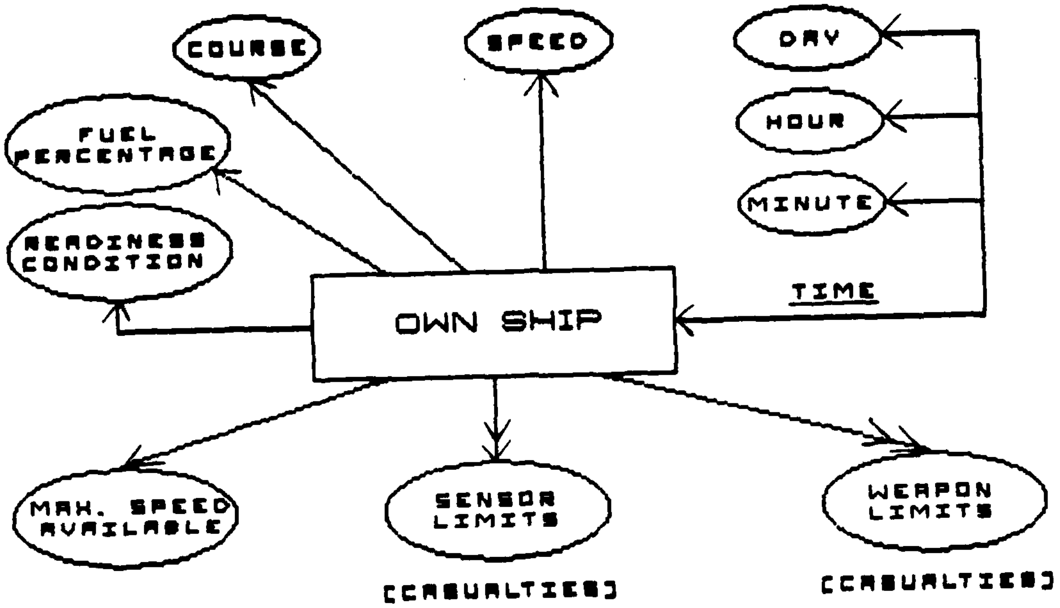


Figure 3.19 ERD for OWN SHIP

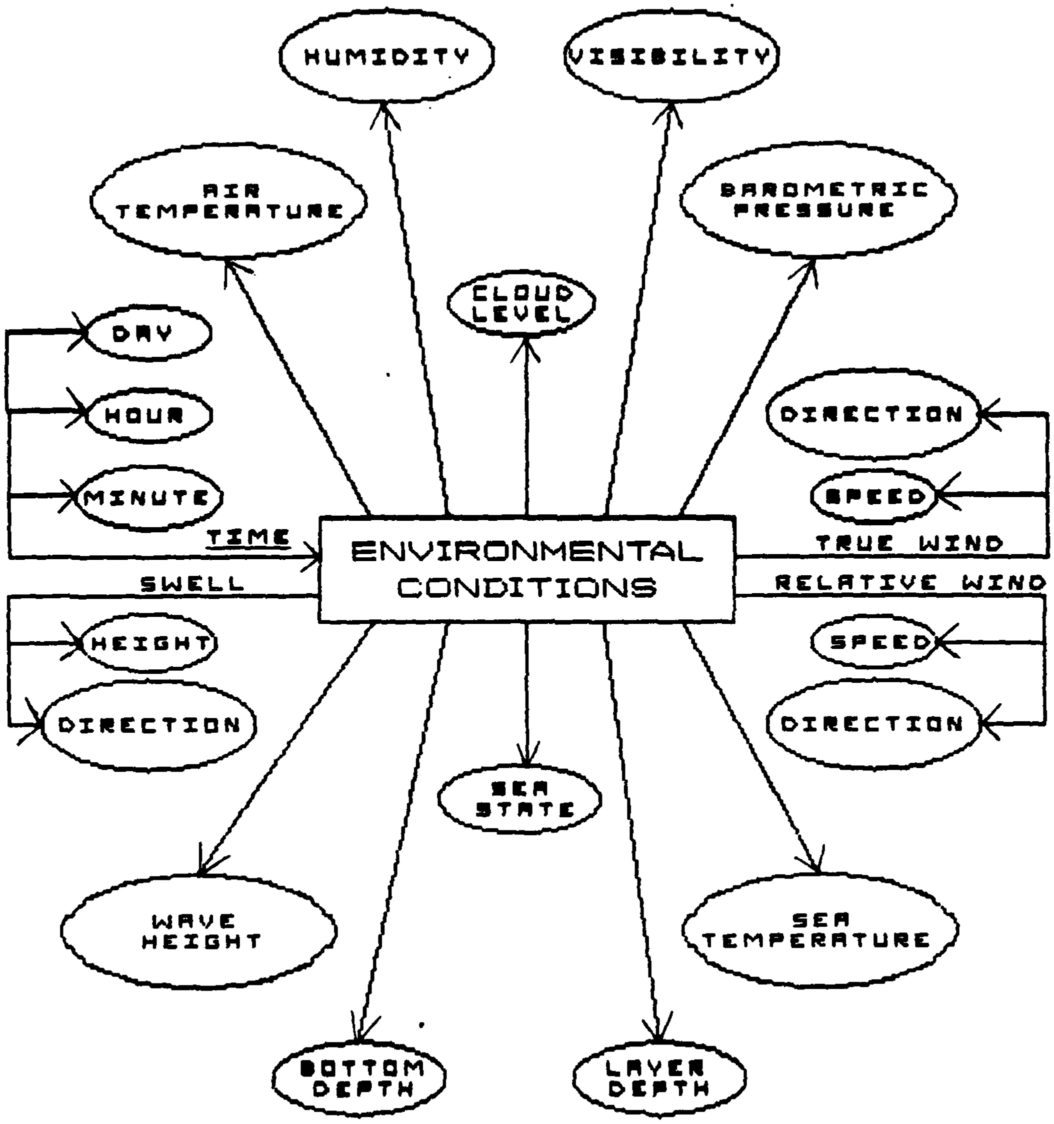


Figure 3.20 ERD for ENVIRONMENTAL CONDITIONS

are given relative to true north and not relative to the ship's head, course must be used to calculate masking conditions. Readiness condition documents the ship's preparedness for battle (general quarters, peacetime steaming, etc.), and the attributes of OWN SHIP that involve sensor and weapon casualties are important because they allow the modeling of limitations to sensors and weapons due to equipment malfunctions/repairs.

ENVIRONMENTAL CONDITIONS would have attributes that capture the important features of the environmental state at a point in time. This information is certainly important in a tactical situation because sensors and weapon systems are limited by the values of these attributes. For example, targeting of guns (ballistic projectiles) must take into account the humidity, temperature, barometric pressure and wind, and the optimum operation of sonars requires consideration of waves, swells, bottom depth, etc. Other attributes that help describe the prevailing environment can be added for a robust TDS.

OWN SHIP and ENVIRONMENTAL CONDITIONS each have associations with SENSOR and WEAPON because the values of some attributes will determine limits on the performance of weapons and sensors. Figure 3.21 shows how these final two entity sets can be related to SENSOR and WEAPON by the use of four new relationship sets: E.C. LIMITS ON S., E.C. LIMITS ON W., O.S. LIMITS ON S., and O.S. LIMITS ON W.

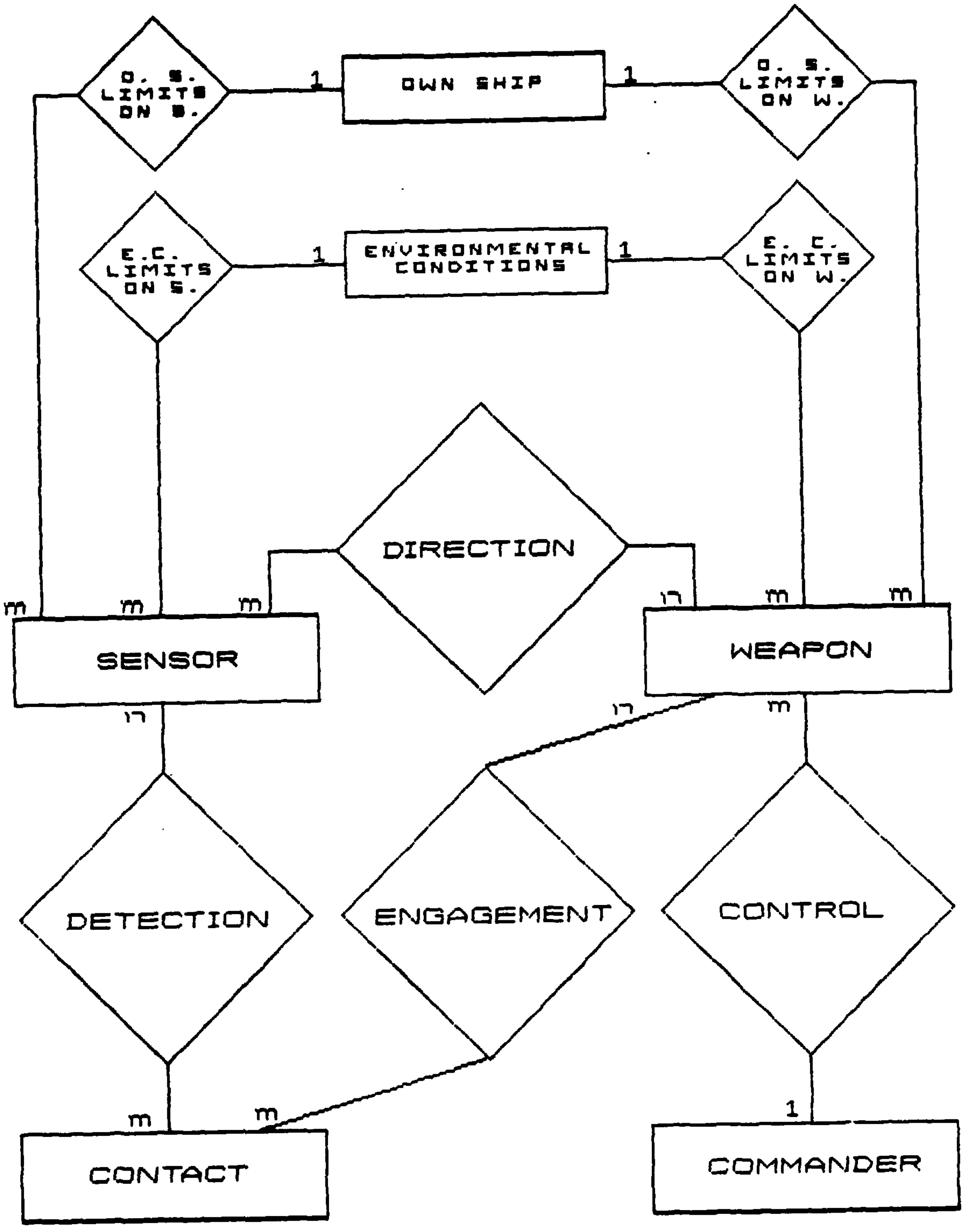


Figure 3.21 ERD Showing all Entity/Relationship Sets

This completes the presentation of the basic conceptual schema (logical database design), but before behavior modeling is considered, a modest extension of the ER diagrammatic technique will be discussed. Figure 3.22 shows the basic high level ERD (less OWN SHIP, ENVIRONMENTAL CONDITIONS and the relationship sets they generate) in a diagram that incorporates the lower levels. This ERD uses a convention that eliminates the need to show the ISA relationship sets, because they are assumed wherever one box (entity set) is contained in another box. Furthermore, relationship sets between sub-entities can be shown. For example, sub-surface contacts can be detected by any of the six sub-entity sets of SENSOR, but surface contacts can only be detected by five and air contacts by only four of them.

D. BEHAVIOR

Since a contact entity in a tactical data system goes through a kind of birth-life-death process, it may be useful to develop a means to model the TDS in a way that will allow these stages to be described. Other entity and relationship tuples also display different behavior over time. Two approaches were considered for this study, and one approach was attempted. The results are discussed in this section.

Ferg [Ref. 6] presents a technique that was used for the Banking Statistics project of the Federal Reserve Board. Ferg's technique is to create an additional entity set for

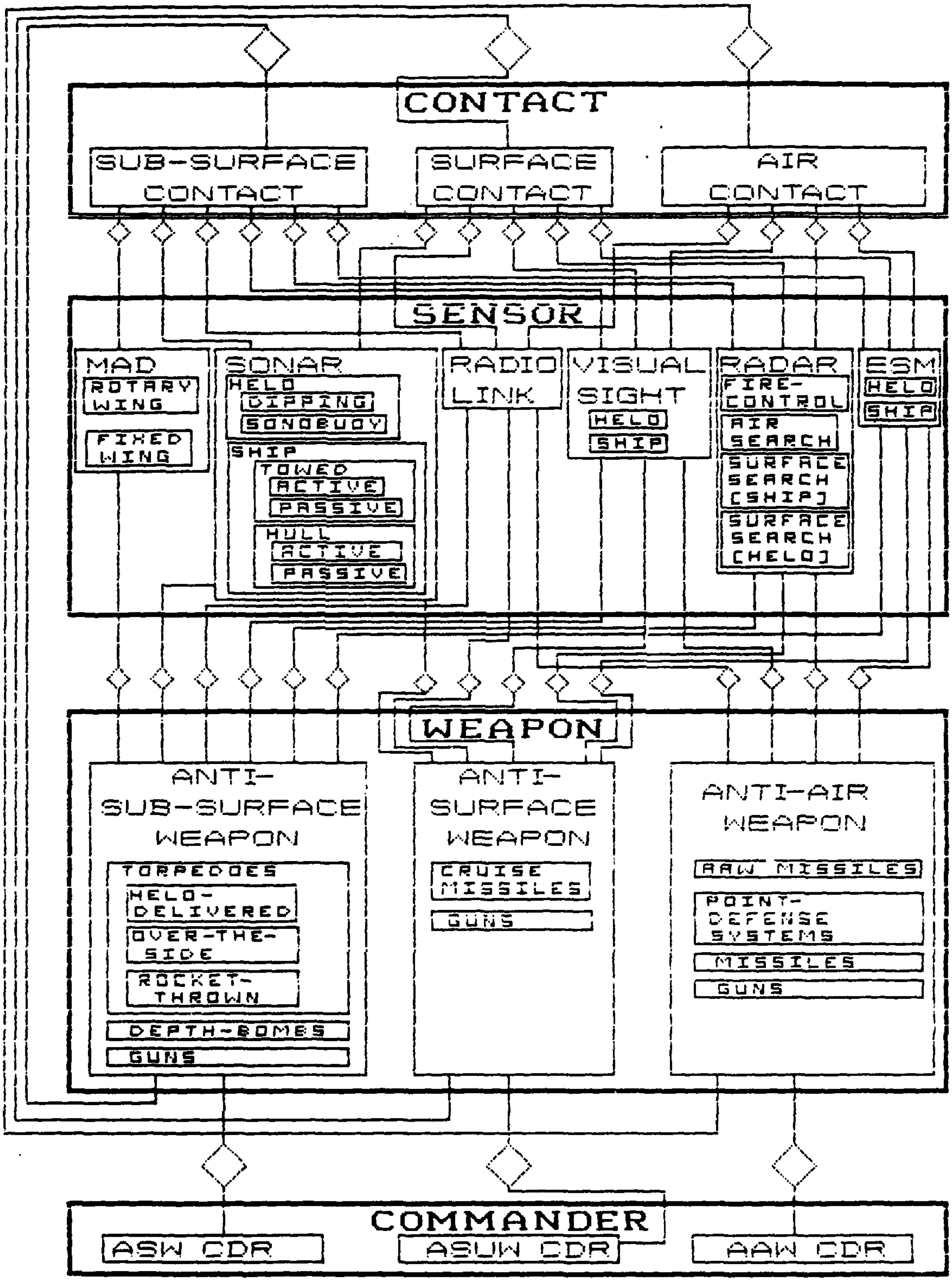


Figure 3.22   ERD (Extension) Basic E/R Sets

every relationship set and think of it as the time period of that relationship set. The two attributes of the time period are functions to time values, and represent the beginning and ending times for the relationship. These "timestamps" thus give the history of the relationship between two entities. This technique should work nicely on relationships such as DETECTION, DIRECTION, ENGAGEMENT and CONTROL and would be relatively simple to add to the schema.

Sakai and Horiuchi [Ref. 7] proposed the use of Petri nets to describe behavior and thus fill out the conceptual schema with the modeling of the time dimension. A Petri net has four basic elements: places (or states), transitions, arcs, and tokens. Figure 3.23 is a Petri net that could be used to describe the behavior of tuples in the CONTACT entity set. Places are represented by elipses, transitions are represented by vertical lines, and arcs are represented as lines with arrow heads. Imagine tokens to be small discs that inhabit the places; then the tokens could describe the "state" that the system is in at any moment in time. A tuple in CONTACT always begins with a token in the /NIL/ place. A transition is enabled if there is at least one input token in each of its input places, and when a transition is enabled, it may fire which causes one token to be removed from each input place and one token to be deposited in each output place. In Figure 3.23, the transition labeled "sensor gains contact" can fire if there is at least

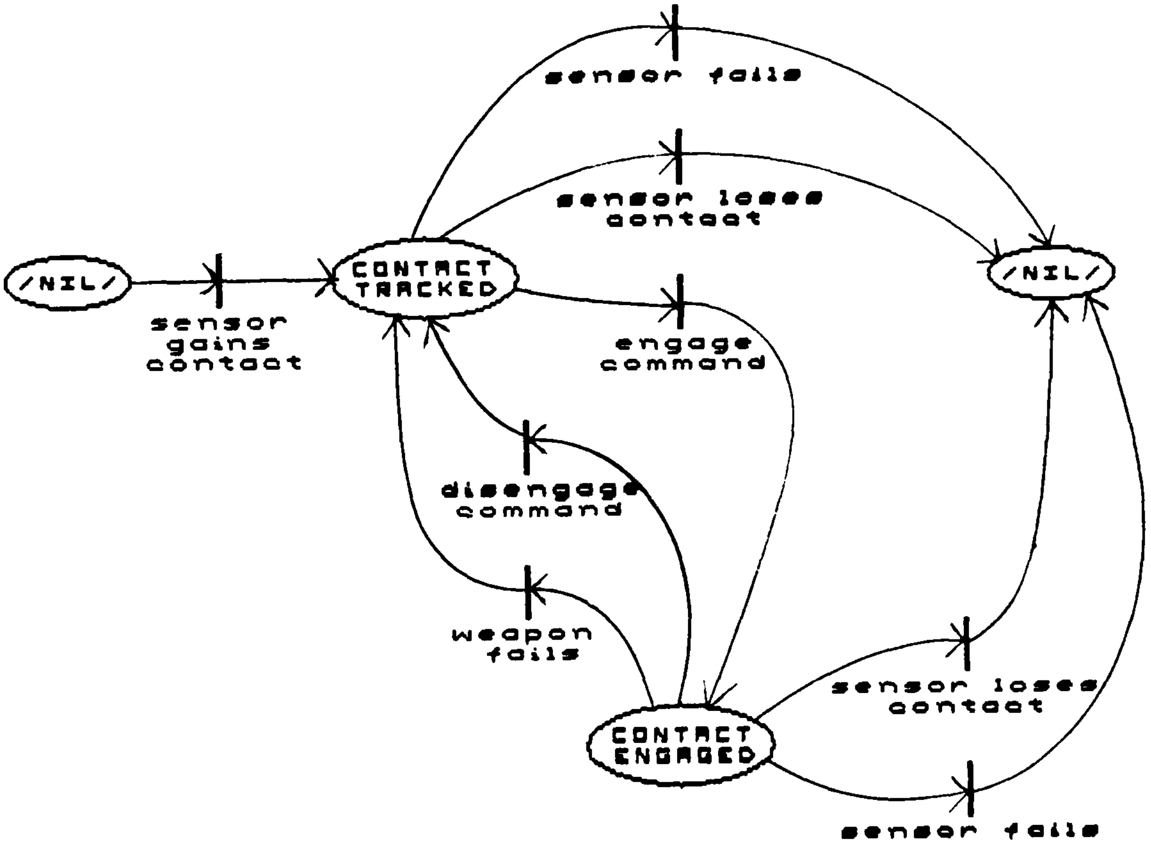


Figure 3.23 Petri Net for CONTACT Entity Set

one token in the input place labeled "/NIL/". The transition labeled "engage command" can only fire if there is a token in the place labeled "CONTACT TRACKED" and so on. So it can be said that the places in Figure 3.23 represent different states in which a contact can be during its lifetime, with the current position of tokens descriptive of the contact's state at *any given moment.* Behavior analysis of each entity/relationship set could yield Petri nets more complex than Figure 3.23 depending on the level of abstraction that is deemed necessary for the application, but clearly a Petri net can be constructed to model the basic behavior.

Figures 3.24 through 3.28 show Petri nets for behavior descriptions of the other basic entity/relationship sets of the conceptual schema presented in earlier sections of this chapter. The technique suggested by Sakai and Horiuchi is to create one integrated Petri net from the individual Petri nets of an ER diagram, and then go through a normalization process which is described in the paper [Ref. 7]. The final resulting Petri net models the behavior of each entity/relationship set and stands as an extension to the ER diagram. An attempt was made to integrate Figures 3.23 through 3.28 and the result was a spaghetti-like net that was more confusing than enlightening. The key conclusion drawn, however, was not that the integration was a bad idea because of the confusing diagram, but that the integration was a bad

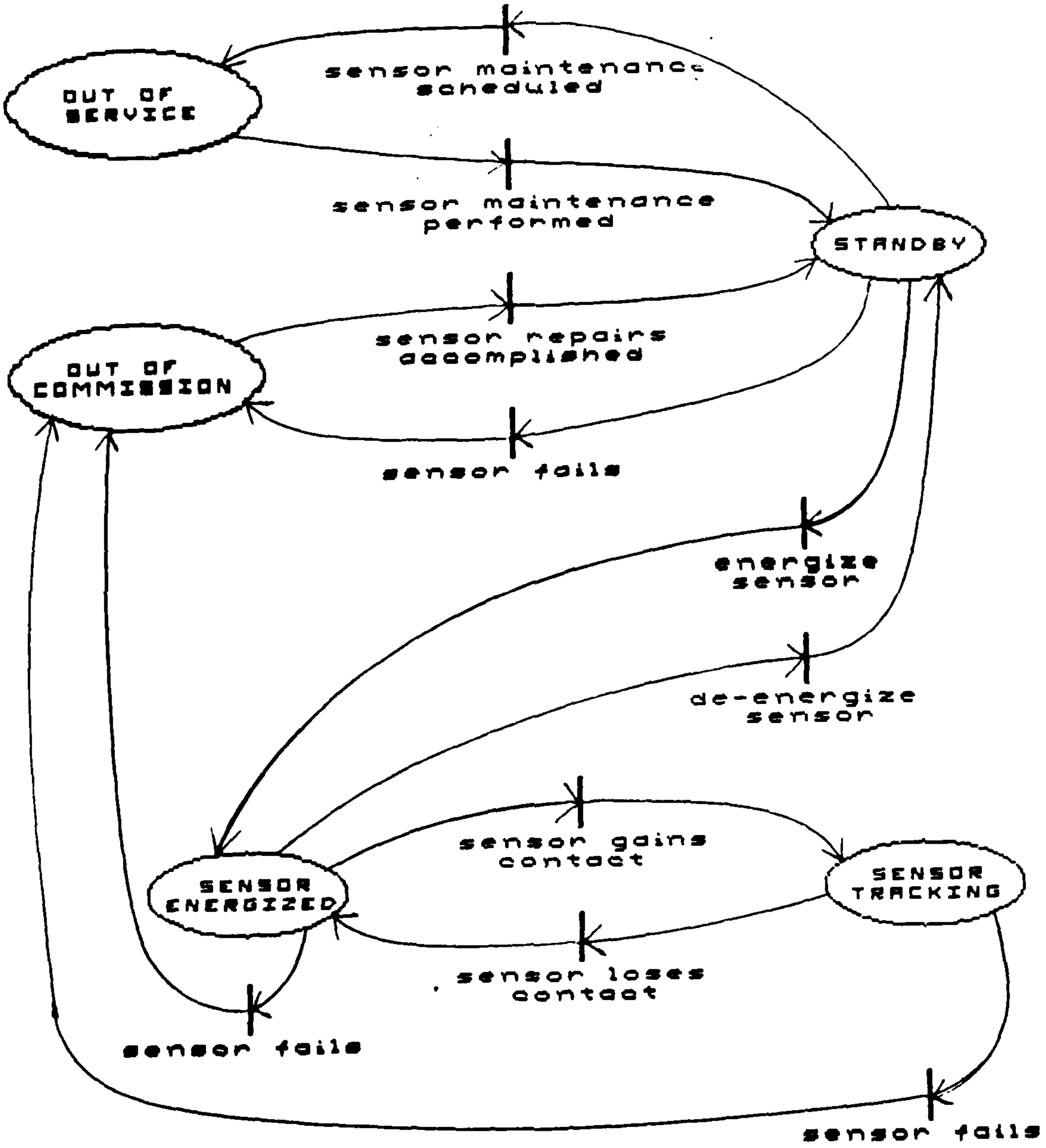


Figure 3.24 Petri Net for SENSOR Entity Set

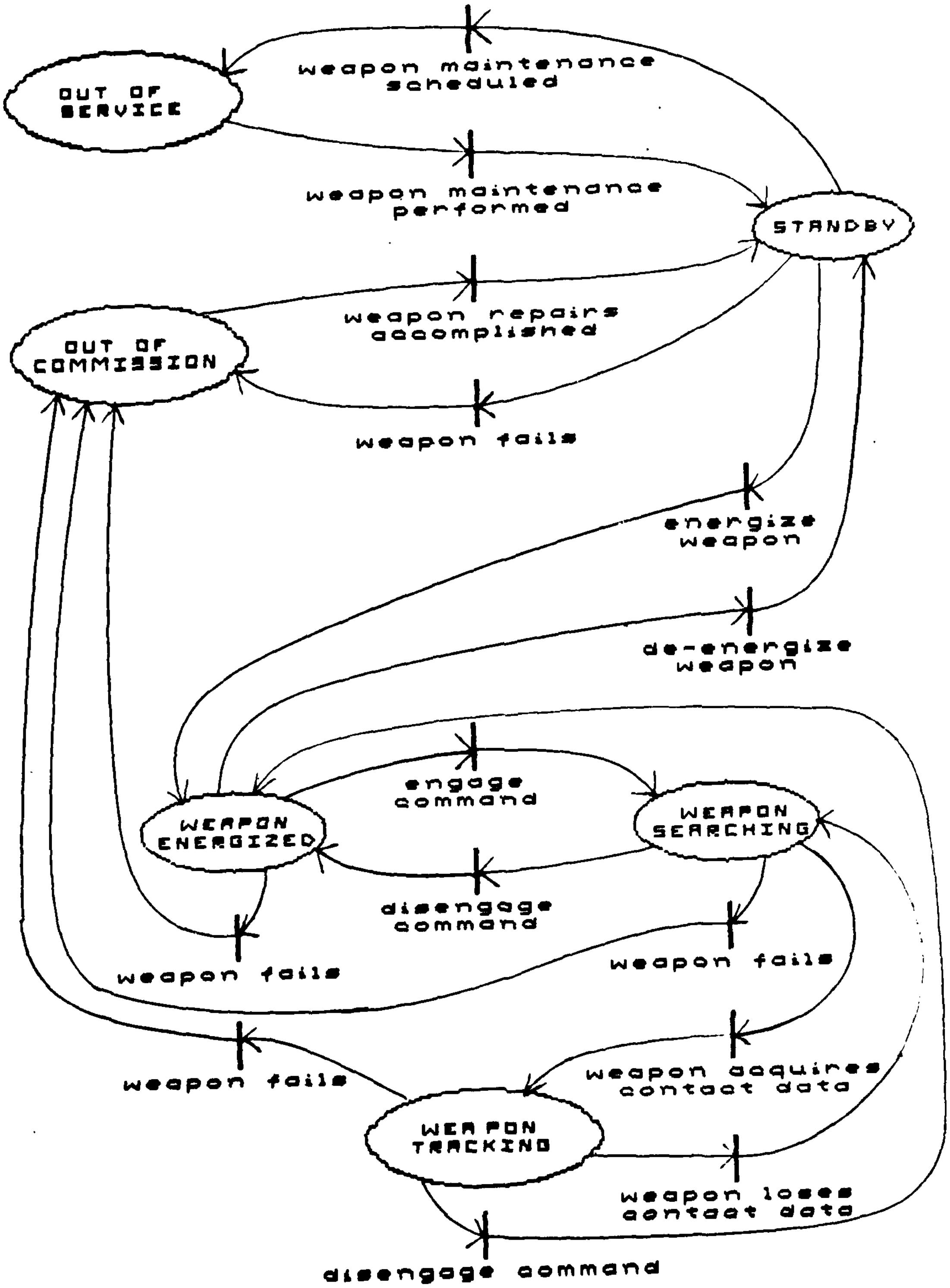


Figure 3.25 Petri Net for WEAPON Entity Set

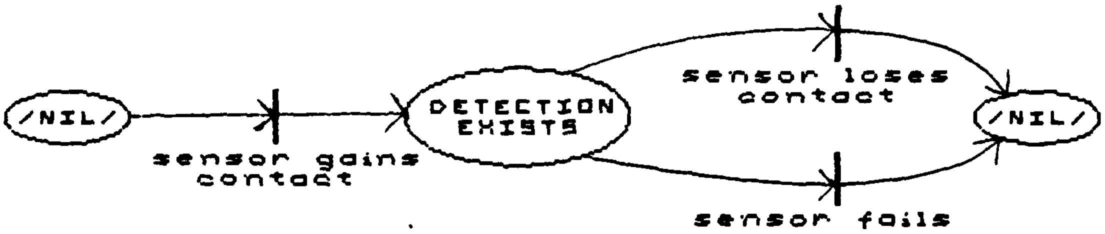


Figure 3.26 Petri Net for DETECTION Relationship Set

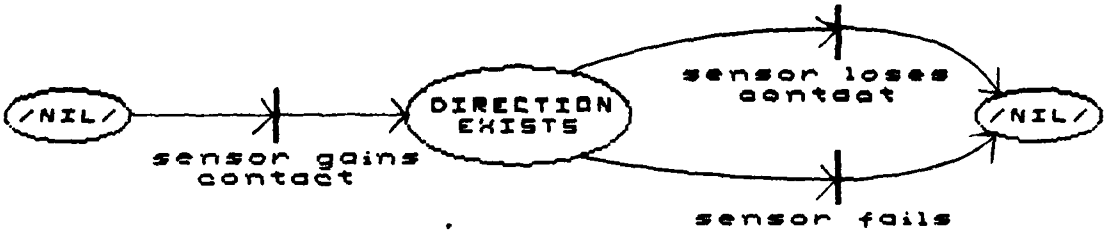


Figure 3.27 Petri Net for DIRECTION Relationship Set

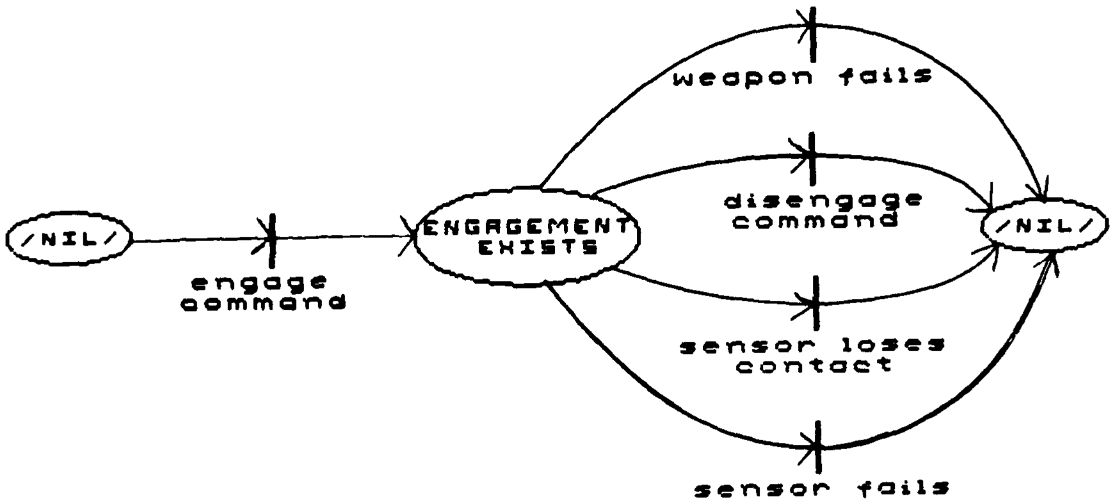


Figure 3.28 Petri Net for ENGAGEMENT Relationship Set

idea because of the inflexibility of the diagram. If, for example, a completely integrated Petri net was completed for a TDS conceptual schema, and then it was determined that new entity sets were needed, the modification of the schema would be fairly straight forward but the modification of the associated integrated Petri net would be nothing short of intimidating. This problem would undoubtably cause the integrated Petri net to fall into disuse.

Even though the schema-wide Petri net turns out to be clumsy and inelegant as a behavior modeling tool, the individual entity/relationship set Petri nets, such as those shown in Figures 3.23 through 3.28, can be useful, primarily as guides during the design of applications that would run over the database. The Sakai and Horiuchi technique attempts to extend the ER approach by appending a large Petri net and its associated state and transition descriptions to the conceptual schema. For small schemas (i.e., those with few entity/relationship sets) the technique would probably work well, but because a TDS is more complex and the logical design needs to be flexible, it appears that it is best to apply a truncated version of the technique (i.e., stop short of integration). The Petri nets seem to find their best use as something more akin to design and maintenance tools than to behavior modeling.

IV. CONCLUSIONS

The Entity-Relationship approach seems to be nicely suited to model the logical database design (conceptual schema) for a TDS. The schema presented in Chapter III is evidence that the ER approach can be applied successfully to the design of a logical database for a TDS. Although questions may persist as to whether the ER model is the best model to use for a TDS, it certainly is a workable model.

The strongest argument for using the ER approach for this type of conceptual schema is its underlying simplicity. Most database models are unnatural to use for laymen who are unfamiliar with database management system issues. But it is the layman who is precisely the one who must validate the design, since he/she is the domain expert and understands best the semantics of the real-world situation which is modeled. Surely a conceptual schema which depicts the real-world situation in the simplest possible way is preferable to one that is more difficult to understand, all other things being equal. It is one contention of this thesis that the ER approach results in schema designs that are easily understood yet powerful and unambiguous.

Flexibility is another issue that has been addressed here, and that is because the typical TDS of the future will have to adjust to dramatic changes in weaponry, sensors, tactics, and even command structures over its deployed lifetime. Conceptual schemas based on the ER approach are relatively easy to modify. For instance, the overall structure of the design presented in Chapter III would not have to be changed to accommodate new weaponry, even if the new weaponry were functionally different from those weapons already incorporated. To accommodate a weapon designed to shoot down satellites orbiting the earth, a new functional WEAPON sub-entity set could be added and named ASPAW WEAPON for anti-space warfare weapon. This shows that the conceptual schema of Chapter III is generic and its overall structure can be ported to organize databases for similar but different TDS problems. Different schemas designed by others using the ER approach might also be generic in this sense.

It seems desirable for a TDS logical database design to have the behavior of entities and relationships modeled over time. Ferg [Ref. 6] has shown one relatively simple way in which this can be done, and his technique could be applied to the logical design presented here. The Sakai and Horiuchi technique [Ref. 7] of developing a large integrated and normalized Petri net to model behavior would not produce a very flexible extension to the ER model when used to model a TDS. Despite the fact that the schema-wide normalized Petri net is intimidating and probably would fall into disuse, individual Petri nets describing the behavior of each entity/relationship set can act as guides to logical database designers, application designers, and maintenance programmers. For this reason, and because the nets are easily developed, they should be considered as additions to the logical database design products for a TDS.

Future research can be directed to the building of new logical database designs using other semantic models with a view to comparing the designs with the one presented in Chapter III. If one particular semantic model proves to be best for TDS conceptual schemas, the design constructed using that model should be filled out to a completely robust stage and finally the conceptual schema should be translated into an internal schema and actual application programs should be designed and written for the database. Once TDS applications can be tested over logical database designs, measurements of speed can be taken. Speed is a significant issue facing those at NOSC now contemplating the feasibility of using DBM techniques for future TDS systems.

It may be shown that TDS speed requirements preclude the use of DBM techniques with current hardware technology, but these systems will be necessary for the Navy for decades to come, and it seems plausible to expect that eventual use of DBM ideas will become reality. It would seem that the Entity-Relationship model provides a good workable approach to designing the conceptual view for the tactical data system of the future.

LIST OF REFERENCES

1. Chen, P. P., "The Entity-Relationship Model--Toward a Unified View of Data," ACM Transactions of Database Systems, Vol.1, No.1, March 1976.

2. Chen, P. P., "The Entity-Relationship Model--A Basis for the Enterprise View of Data," AFIPS Conference Proceedings 1977 National Computer Conference, AFIPS Press, 1977.

3. ANSI/X3/SPARC, "Interim Report from the Study Group on Data Base Management Systems," Bulletin of the ACM SIGMOD, Vol. 7, No. 2, 1975.

4. Clemons, E. K., "Data Models and the ANSI/SPARC Architecture," Principles of Database Design: Volume I Logical Organizations, Prentice-Hall, Inc., 1985.

5. Chen, P. P., "Database Design Based on Entity and Relationship," Principles of Database Design: Volume I Logical Organizations, Prentice-Hall, Inc., 1985.

6. Ferg, S., "Modelling the Time Dimension in an Entity-Relationship Diagram," Proceedings: The 4th International Conference on Entity-Relationship Approach, IEEE Computer Society Press, 1985.

7. Sakai, H. and Horiuchi, H., "A Method for Behavior Modeling in Data Oriented Approach to Systems Design," IEEE Data Engineering, 1984.

BIBLIOGRAPHY

ANSI/X3/SPARC, "Interim Report from the Study Group on Data Base Management Systems," Bulletin of the ACM SIGMOD, Vol. 7, No. 2, 1975.

Chen, P. P., "The Entity-Relationship Model--Toward a Unified View of Data," ACM Transactions of Database Systems, Vol.1, No.1, March 1976.

Chen, P. P., "The Entity-Relationship Model--A Basis for the Enterprise View of Data," AFIPS Conference Proceedings 1977 National Computer Conference, AFIPS Press, 1977.

Chen, P. P. and Dogac, A., "Entity-Relationship Model in the ANSI/SPARC Framework," Entity-Relationship Approach to Information Modeling and Analysis, ER Institute, 1981.

Chen, P. P., "Database Design Based on Entity and Relationship," Principles of Database Design: Volume I Logical Organizations, Prentice-Hall, Inc., 1985.

Clemons, E. K., "Data Models and the ANSI/SPARC Architecture," Principles of Database Design: Volume I Logical Organizations, Prentice-Hall, Inc., 1985.

Ferg, S. "Modelling the Time Dimension in an Entity-Relationship Diagram," Proceedings: The 4th International Conference on Entity-Relationship Approach, IEEE Computer Society Press, 1985.

Sakai, H. and Horiuchi, H., "A Method for Behavior Modeling in Data Oriented Approach to Systems Design," IEEE Data Engineering, 1984.

Smith, J. M. and Smith, D. C. P., "Database Abstractions: Aggregation and Generalization," ACM Transactions on Database Systems, Vol 2., No. 2., June 1977.

Ullman, J. D., Principles of Database Systems, Computer Science Press, 1982.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22304-6145

2. Library, Code 0142 2
Naval Postgraduate School
Monterey, California 93943-5000

3. Department Chairman, Code 52 2
Department of Computer Science
Naval Postgraduate School
Monterey, California 93943-5000

4. Curriculum Officer, Code 37 1
Computer Technology
Naval Postgraduate School
Monterey, California 93943-5000

5. Professor C. Thomas Wu, Code 52WQ 2
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5000

6. Professor David K. Hsiao, Code 52HQ 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5000

7. Professor Peter P. Chen 1
Murphy J. Foster Distinguished Chair Professor
Department of Computer Science
Louisiana State University
Baton Rouge, LA 70803-4020

8. Naval Ocean Systems Center 3
Attn: L. Y. Wong, Code 443
San Diego, California 92152

9. LT R. P. Kauffold, USN 2
3428 Flint Hill Place
Woodbridge, Virginia 22192

END

10-86

DTIC